LEVEL II

12

AD A108682

R and D CENTER LABORATORY

TECHNICAL REPORT

NO. 12589

KINEMATIC ANALYSIS OF MECHANICAL SYSTEMS

Interim Report

16 June 1981

Contract No. DAAK30-78-C-0096

DTIC ELECTE DEC 17 1981

Rajiv Rampalli & Edward J. Haug
College of Engineering
The University of Iowa
University of Iowa Rept. No. 60

by Ronald R. Beck, Project Engineer, TACOM

Approved for public release; distribution unlimited.

DTIC FILE COPY

U.S. ARMY TANK-AUTOMOTIVE COMMAND
RESEARCH AND DEVELOPMENT CENTER
Warren, Michigan 48090

81 12 17 068

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

NOTICES

The findings in this report are not to be construed as an official Department of the Army position.

Mention of any trade names or manufacturers in this report shall not be construed as advertising nor as an official endorsement of approval of such products or companies by the U.S. Government.

Destroy this report when it is no longer needed. Do not return it to originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 12589	2. GOVT ACCESSION NO. AD-A108 682	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Kinematic Analysis of Mechanical Systems		5. TYPE OF REPORT & PERIOD COVERED Interim to Dec. 79
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Rajiv Rampalli & Edward J. Haug University of Iowa Ronald R. Beck, TACOM		8. CONTRACT OR GRANT NUMBER(s) DAAK30-78-C-0096
9. PERFORMING ORGANIZATION NAME AND ADDRESS The University of Iowa College of Engineering Iowa City, IA 52242		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Tank-Automotive Command R&D Center Tank-Automotive Concepts Lab, DRSTA-ZSA Warren, MI 48090		12. REPORT DATE 16 June 1981 13. NUMBER OF PAGES 65
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Dynamics, Bodies, Response, Control, Interaction, Stability, Lagrangian Function, Matrices

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

It is intended in this report to develop a method of kinematically analyzing mechanical systems. A computer code that automatically generates a set of nonlinear equations describing the system and solves them, is developed. To demonstrate the accuracy and effectiveness of the method to be developed herein, five sample problems have been solved.

While all the sample problems solved are closed-loop mechanical systems, the method described here need not be confined to closed loop mechanical

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

4/0 5316

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

systems or to mechanical systems at all. In fact, the method is applicable to any system whose motion is to be described, provided adequate constraints can be incorporated into the code by the user. Examples of such systems are electro-mechanical systems with relays, switches, etc.

Given the topology of a system, the necessary inputs, the method is capable of computing velocities, positions, accelerations and constraint forces on the system.

Briefly, the most notable features of this method are:

(i) Complete generality in the derivation of the constraint equations and computation of the generalized co-ordinates.

(ii) Use of sparse matrix subroutines to realize substantial savings in core when problems of large magnitude are analyzed.

(iii) Versatility to incorporate nonstandard constraints into a problem, if supplied by the user.

(iv) Ability to handle constraints, when they are supplied as a set of discrete points, by constructing a third-order spline function through them.

(v) Ability to perform kineto-static analysis of a mechanical system if the system is kinematically indeterminate.

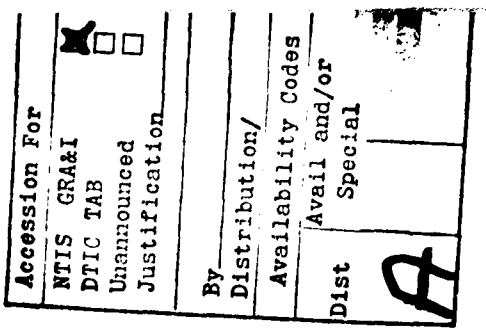


TABLE OF CONTENTS

INTRODUCTION

It is intended in this report to develop a method of kinematically analyzing mechanical systems. A computer code that automatically generates a set of nonlinear equations describing the system and solves them, is developed. To demonstrate the accuracy and effectiveness of the method to be developed herein, five sample problems have been solved.

While all the sample problems solved are closed-loop mechanical systems, the method described here need not be confined to closed loop mechanical systems or to mechanical systems at all. In fact, the method is applicable to any system whose motion is to be described, provided adequate constraints can be incorporated into the code by the user. Examples of such systems are electro-mechanical systems with relays, switches, etc.

Given the topology of a system, and the necessary inputs, the method is capable of computing velocities, positions, accelerations and constraint forces on the system.

Briefly, the most notable features of this method are:

(i) Complete generality in the derivation of the constraint equations and computation of the generalized co-ordinates.

(ii) Use of sparse matrix subroutines to realize substantial savings in core when problems of large magnitude are analyzed.

(iii) Versatility to incorporate nonstandard constraints into a problem, if supplied by the user.

(iv) Ability to handle constraints, when they are supplied as a set of discrete points, by constructing a third-order spline function through them.

(v) Ability to perform kineto-static analysis of a mechanical system if the system is kinematically indeterminate.

LIST OF SYMBOLS

α	Imput parameter for kinematically determinate systems
δ	Symbols for virtual variations
λ	Lagrange multipliers corresponding to constraints
ξ,η	x, y axes of body-fixed coordinate systems
ϕ	Vector of constraint equations
θ	Generalized coordinate corresponding to rotation
f,g,g_1	Vector of equations
r_{ij},r_{ji},r_p	Vectors used in defining constraint equations
R	Spring constant for linear springs
Rr	Spring constant for torsional springs
ℓ	Deformed length of springs
ℓ_0	Free length of spring
I	Identity matrix
L	Lower triangular factor of the Jacobian matrix
Q,F	Vector of applied forces
(T_{ij})	Rotation matrix between references i and j
U	Upper triangular factor of the Jacobian matrix
X,Y	Translational generalized coordinates of a body
KE	Kinetic energy of the mechanical system

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1.1 Introduction:

The objective of this report is to formulate and implement a computer code that is capable of performing kinematic analysis of constrained mechanical systems. The need for such a code arose because most of the available codes at present, for example IMP, ADAMS, and DADS, are optimized for transient dynamic analysis, rather than for kinematic analysis. Using the above codes to perform kinematic analysis is tantamount to underutilization of the code. Not surprisingly, therefore, the analysis process becomes unnecessarily expensive and time consuming.

In this report, the DADS-2D* code [1] is modified to do kinematic analysis. The discussion here is limited to two dimensional problems, but the method is applicable to more general three dimensional systems, with few modifications.

1.2 Statement of the problem:

Given kinematic or force inputs to a mechanical system of known topology, one wishes to solve the equations of equilibrium to obtain an equilibrium position of the system that is consistent with its constraints.

It is assumed that necessary matrix methods and other numerical algorithms are available so they will not be discussed in detail here.

1.3 Literature Review:

Several general-purpose computer programs for dynamic and kinematic analysis of mechanisms have been described in the literature. These programs, which automatically formulate and integrate the equations of motion are based on different but related analytical and numerical principles.

The major general-purpose programs available at the time of writing are given in the following list (also see lists in Refs. 3, 4, and 8):

*2D implies a code that is capable of analyzing planar systems only.

Also, refer to [4] for further details.

IMP (Integrated Mechanisms Program), described by Sheth and Uicker [5];

DYMAC (Dynamics of Machinery), developed by Burton Paul, G. Hud and A. Amin at the University of Pennsylvania [6];

DRAM (Dynamic Response of Articulated Machinery), described by Chace and Smith [7];

MEDUSA (Machine Dynamics Universal System Analyzer) described by Dix and Lehman [8];

ADAMS (Automated Dynamic Analysis of Mechanical Systems), developed at the University of Michigan by Orlandea et al. is described in Orlandea and Chace [9];

DADS-2D (Dynamic Analysis of Dynamic Systems), described by R.A. Wehage, E.J. Haug, Jr., and R.C. Huang [1, 10];

DADS-3D (Dynamic Analysis of Dynamic Systems), described by R.C. Huang and E.J. Haug, Jr. [1];

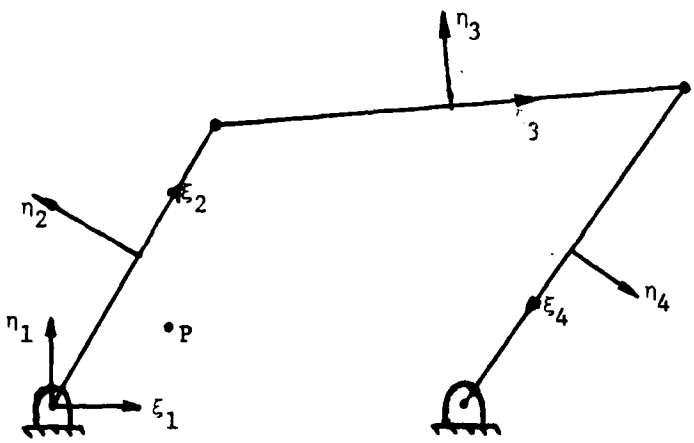
In addition to the above, there are special purpose programs that can analyze only open loop systems. These include:

AAPD (Automated Assembly Program), a program for 3 dimensional mechanisms, is described by Langrana and Bartel [11]. Only revolute and spherical pairs are allowed.

UCIN, developed by Houston et al. at the University of Cincinnati, is for strictly open loops [12].

Loop Closure Technique

Loop closure techniques essentially make use of the fact that the product of transformation matrices around a closed loop is the identity matrix. Consider, for instance, a simple 4-bar mechanism.



Four Bar Mechanism

Fig. 1.1

Consider any point of interest P, whose coordinates are to be determined in the various frames of reference (ξ_1, η_1), (ξ_2, η_2), (ξ_3, η_3), (ξ_4, η_4), the ξ axes of which are aligned along the length of the respective bodies at each joint.

In reference frame (ξ_1, η_1) point P has the coordinates

$$[P]_1 = \begin{bmatrix} \xi_{p1} \\ \eta_{p1} \end{bmatrix} \tag{1.3.1}$$

(The subscripts on ξ_p and η_p denote the frame of reference).

In reference frame (ξ_2, η_2)

$$[P]_2 = (T_{12})[P_1] = (T_{12}) \begin{bmatrix} \xi_{p1} \\ \eta_{p1} \end{bmatrix} \tag{1.3.2}$$

where (T_{12}) is the rotation matrix relating coordinates (ξ_1, η_1) and (ξ_2, η_2).

Similarly, in reference frame (ξ_3, η_3),

$$[P]_3 = \begin{bmatrix} \xi_{p3} \\ \eta_{p3} \end{bmatrix} = (T_{23}) \begin{bmatrix} \xi_{p2} \\ \eta_{p2} \end{bmatrix} \tag{1.3.3}$$

where (T_{23}) is the rotational matrix from reference frames 2 to 3.

In reference frame (ξ_4, η_4)

$$[P]_4 = \begin{bmatrix} \xi_{p4} \\ \eta_{p4} \end{bmatrix} = (T_{34}) \begin{bmatrix} \xi_{p3} \\ \eta_{p3} \end{bmatrix} \tag{1.3.4}$$

where (T_{34}) is the rotational matrix from reference frames 3 to 4.

Finally,

$$[P]_1 = \begin{bmatrix} \xi_{p1} \\ \xi_{p1} \end{bmatrix} = (T_{41}) \begin{bmatrix} \xi_{p4} \\ \eta_{p4} \end{bmatrix} \tag{1.3.5}$$

where (T_{41}) is the rotational matrix from reference frames 4 to 1.

From equations (1.3.1) to (1.3.5), one has

$$[P]_1 = (T_{41})(T_{34})(T_{23})(T_{12})\ [P]_1 \tag{1.3.6}$$

Since P is chosen arbitrarily, (1.3.6) implies that

$$(T_{41})(T_{34})(T_{23})(T_{12}) = [I] \tag{1.3.7}$$

Considering the first variation of equation (1.3.7), an iterative technique can be developed. Beginning with an initial estimate of the angles and positions of each joint, a solution can be generated such that (1.3.7) holds for each independent loop in the system.

There are three basic disadvantages in using the loop closure:

(a) The method is applicable only if a closed loop exists. If there are no closed loops, imaginary links and joints must be incorporated into the system, without changing the kinematics of the system. This normally leads to unnecessary complexities in the completion of loops and unnecessary computations as far as the basic problem itself is concerned.

(b) The equations to be solved are highly nonlinear in nature and, even though they are few in number, convergence might be rather slow.

(c) The program needs to have the capability of recognizing independent loops in the mechanism and selecting the best set of loops, to optimize computational errors.

Programs which use the loop closure technique include IMP, DRAM, and DYMAC.

In contrast, programs like ADAMS, DADS-2D, and DADS-3D utilize a sparse matrix approach, to be presented later, which leads to a large number of loosely coupled equations that are to be solved.

These programs are equally efficient with closed and open loop mechanisms.

1.4 Organization of the Report:

Before dealing with the problem at hand, it is essential to know how the DADS -2D code works, so that one may understand precisely what modifications are necessary to use it effectively for kinematic analysis.

Chapter 2 is, therefore, utilized for explaining the fundamentals of this approach. Only the most basic aspects will be discussed here. The reader is referred to Ref. [1] for details of the method.

Chapter 3 deals with the proposed method of analysis of kinematically determinate systems.

Chapter 4 contains a description of the analysis procedure for force driven systems.

Chapter 5 contains the implementation of the method, a discussion of the computer code used, and briefly explains the computational flow in the program.

In Chapter 6, numerical results for several problems, obtained by the above method are discussed, conclusions are presented and promising directionsfor future effort in this field are discussed.

CHAPTER 2

An Overview of the DADS Program

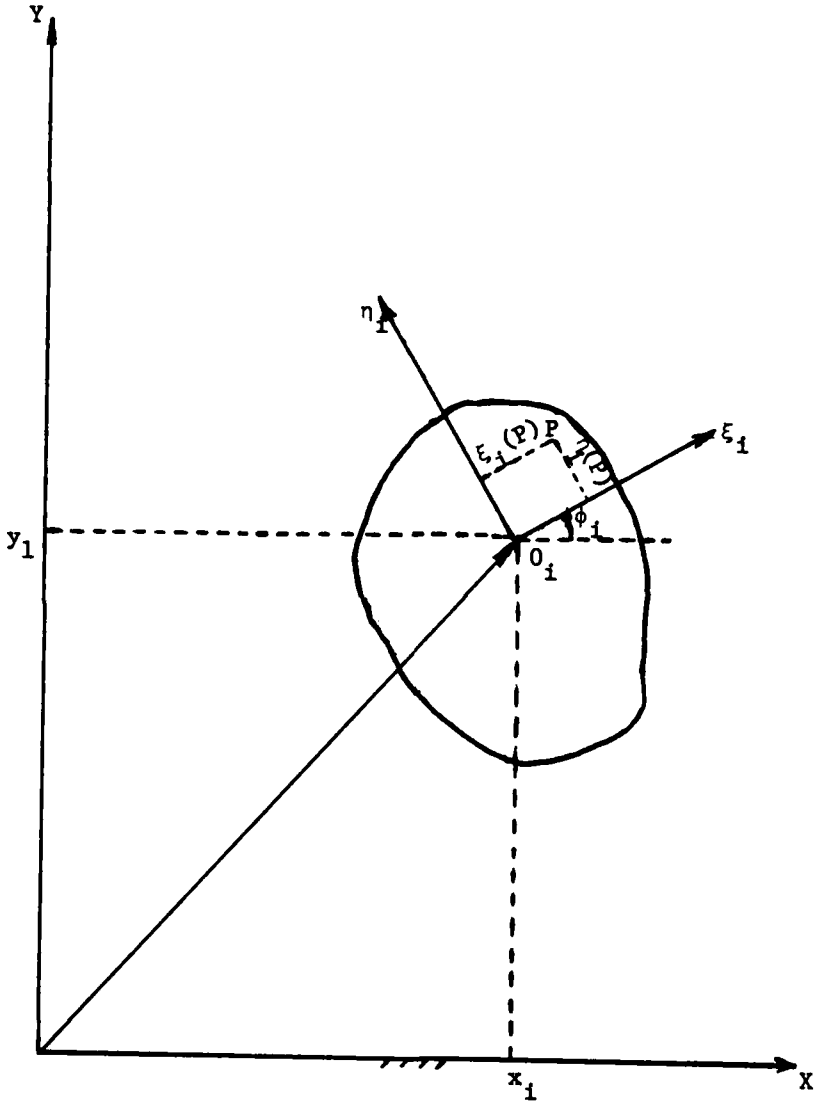
2.1

Any rigid body undergoing planar motion can be uniquely fixed in space by specifying three coordinates with respect to a Newtonian reference frame. The simplest set of suitable coordinates is the Cartesian X,Y coordinate of some point on the body e.g., the center of mass, with respect to the reference frame, and a ϕ coordinate that defines the rotation of any line on the body with respect to its initial time, as shown in Fig. 2.1. Attached to each body is another frame of reference (O_i, ξ_i, η_i) that is used for the purpose of locating fixed points on that body.

A mechanical system consists of several such rigid bodies connected by means of joints and springs. Joints constrain the motion of the body on which they are located and springs introduce internal forces into the system.

Two types of joints, the revolute joint and the translational joint, are included in the formulation [1]. The DADS-2D program [1], however, is versatile enough to allow the user to insert non-standard joints such as cams and other higher pair joints.

Since mathematical equations can be written for each constraint in the system, the problem of dynamic analysis of a constrained mechanical system reduces to solving the equations of motion corresponding to each generalized coordinate, subject to constraints imposed on the motion by the joints. Suppose that there are 'n' bodies in an arbitrary mechanical system and that they are subjected to '2m' constraints. Corresponding to each body, there are three degrees-of-freedom; so in total, there are 3n-generalized



Coordinate Systems

Figure 2.1

coordinates making up the system, denoted by $q \in R^{3n}$. These coordinates are not independent, however, since they must satisfy the equations of constraint

$$\phi(q) = 0 \tag{2.1.1}$$

where $\phi \in R^{2m}$, and 'm' is the total number of joints in the system.

The equations of motion, derived from the variational form of Lagrange's equations [14], can be written as

$$\sum_{i=1}^{3n} \left\{ \frac{d}{dt}\left(\frac{\partial KE}{\partial \dot{q}_i}\right) - \frac{\partial KE}{\partial q_i} - Q_i \right\} \delta q_i = 0 \tag{2.1.2}$$

where Q_i represent all conservative and non-conservative forces acting in the q_i direction.

The equations of motion (2.1.2) must hold for all time and for all virtual displacements δq_i that are consistent with the constraints (2.1.1), i.e.

$$\left(\frac{\partial \phi_i}{\partial q_j}\right)\delta q = 0 \tag{2.1.3}$$

By the Farkas Lemmas of optimization theory, there exist multipliers λ, $\lambda \in R^{2m}$, such that

$$\sum_{i=1}^{3n} \left\{ \frac{d}{dt}\left(\frac{\partial KE}{\partial \dot{q}_i}\right) - \frac{\partial KE}{\partial q_i} - Q_i + \sum_{j=1}^{2m} \frac{\partial \phi_j}{\partial q_i} \lambda_j \right\} \delta q_i = 0 \tag{2.1.4}$$

for all δq_i.

Thus, one obtains the equations

$$\frac{d}{dt}\left(\frac{\partial KE}{\partial \dot{q}_i}\right) - \frac{\partial KE}{\partial q_i} - Q_i + \sum_{j=1}^{2m} \frac{\partial \phi_j}{\partial q_i} \lambda j = 0, \quad i=1,3n \tag{2.1.5}$$

$$\phi_j (q) = 0 \quad j = 1,2m \tag{2.1.6}$$

One therefore has a set of (3n + 2m) equations in (3n + 2m) variables q and λ. This is a coupled set of differential and algebraic equations that must be solved simultaneously.

2.2 Integration of the Equations of Motion:

The equations of equilibrium are second-order, nonlinear, ordinary differential equations, which can be expressed in the form

$$g(q, \dot{q}, \ddot{q}, \lambda, t) = 0, g \in R^{3n} \tag{2.2.1}$$

Introducing a new variable u, such that one has

$$u - \dot{q} = 0 \tag{2.2.2}$$

$$\dot{u} - \ddot{q} = 0 \tag{2.2.3}$$

The equations of motion can now be written as

$$g_1(q, u, \dot{u}, \lambda, t) = 0, \; g_1 \in R^{6n} \tag{2.2.4}$$

which are first-order and more easily adaptable for computer-aided methods of solution. The numerical algorithm used in the DADS-2D program is Gear's Stiff Integration Algorithm. Essentially, this algorithm replaces the time derivatives of the generalized coordinates at each time step by a polynomial approximation, using the values of the variable and its derivative at the present and previous R time steps. Once the replacement is done, the problem simplifies to a system of algebraic equations, which is solved by the Newton-Raphson iteration scheme [1].

2.3 Sparse Matrix Algebra:

One could theoretically solve the above problem by eliminating the dependent generalized coordinates from the system of equations and reducing the size of the system. This technique, however, leads to a greater degree of coupling between the remaining equations, which also tend to become very nonlinear. With the formulation presented in the preceding section, the equations are very loosely coupled because instead of solving a system of $(3n + 2m)$ equations, a system of $(6n + 2m)$ equations is being solved. Therefore, while performing the Newton-Raphson iterations, one is confronted with a large matrix in which most of the elements are zero. The locations of the non-zero elements and their values can be easily determined, once the topology of the system is known. As long as the topology does not change, their locations also do not change.

Sparse matrix algorithms designed to take maximum benefit of the sparsity of the matrices involved can be put to good use here. A substantial saving in core storage is realized, since the values of the nonzero entries and their row and column addresses can be easily stored in three vectors of small dimension.

In the DADS-2D program [1], the Harwell library of sparse matrix subroutines is used. They perform a symbolic LU factorization of the matrices involved using a partial pivoting technique that maintains a high degree of sparsity in the L and U matrices. The variables are then determined by a simple forward and backward substitution procedure.

No effort is made here to explain the intricate computational details involved in the DADS-code; rather the emphasis here is on the use of the solution technique for the kinematic and equilibrium problem.

For a more detailed discussion, the reader is referred to references 1 and 2.

2.4 Equations of Constraint:

Figure 2.2 depicts two adjacent bodies i and j. The origins of their body fixed coordinate systems are located by the vectors $\bar{R}_i$ and $\bar{R}_j$ with respect to the inertial frame of reference. Let an arbitrary point p_{ij} on body i be located by $\bar{r}_{ij}$ and p_{ji} on body j be located by $\bar{r}_{ji}$. These points are, in turn, connected by a vector $\bar{r}_p$. One can write a vector equation beginning at the origin of the inertial reference frame and closing there, to obtain the vector relationship:

$$\bar{R}_i + \bar{r}_{ij} + \bar{r}_p - \bar{r}_{ji} - \bar{R}_j = \bar{0} \tag{2.4.1}$$

The constraint equations for a revolute joint are now obtained by requiring that p_{ij} and p_{ji} coincide. Setting $\bar{r}_p = \bar{0}$ and writing Eq. 2.4.1 in component form, one has

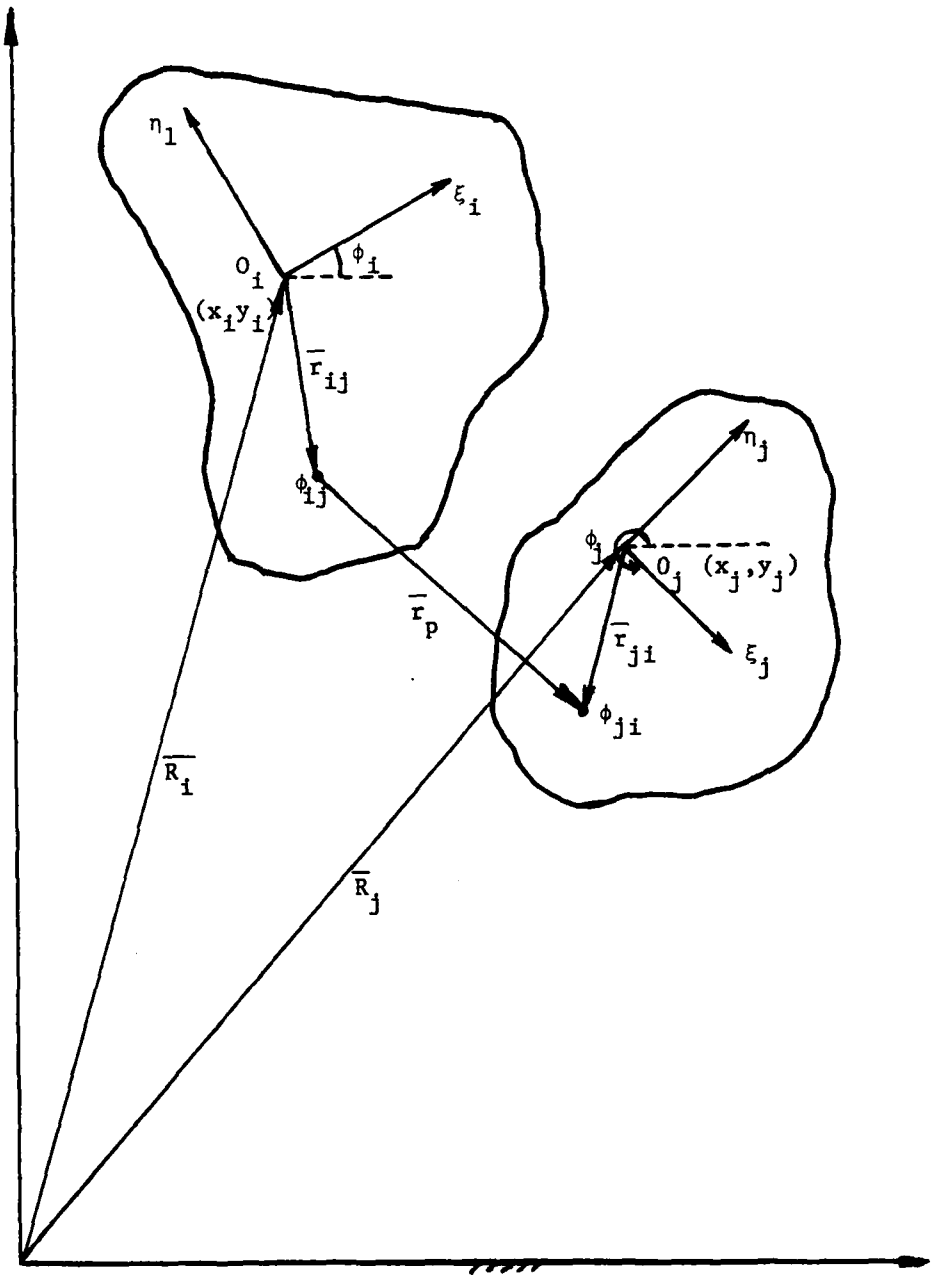
$$\left.\begin{aligned} x_i + \xi_{ij} \cos\phi_i - \eta_{ij} \sin\phi_i - x_j - \xi_{ji} \cos\phi_j + \eta_{ji} \sin\phi_j = 0 \\ y_i + \xi_{ij} \sin\phi_i + \eta_{ij} \cos\phi_i - y_j - \xi_{ji} \sin\phi_j - \eta_{ji} \cos\phi_j = 0 \end{aligned}\right\} \tag{2.4.2}$$

For a translational joint shown in Figure 2.3, let points p_{ij} and p_{ji} lie on some line parallel to the path of relative motion between two bodies. In addition, let them be located such that $\bar{r}_{ij}$ and $\bar{r}_{ji}$ are perpendicular to this line and of nonzero magnitude. Successively forming the dot product of Eq. 2.4.1 with $\bar{r}_{ij}$ and $\bar{r}_{ji}$ and adding, one obtains the scalar equation

$$- \bar{r}_{ij} \cdot \bar{r}_{ji} + r_{ij}^2 + \bar{r}_{ij} \cdot (\bar{R}_i - \bar{R}_j) + \bar{r}_{ji} \cdot \bar{r}_{ij} - r_{ji}^2 + \bar{r}_{ji} \cdot (\bar{R}_i - \bar{R}_j) = 0 \tag{2.4.3}$$

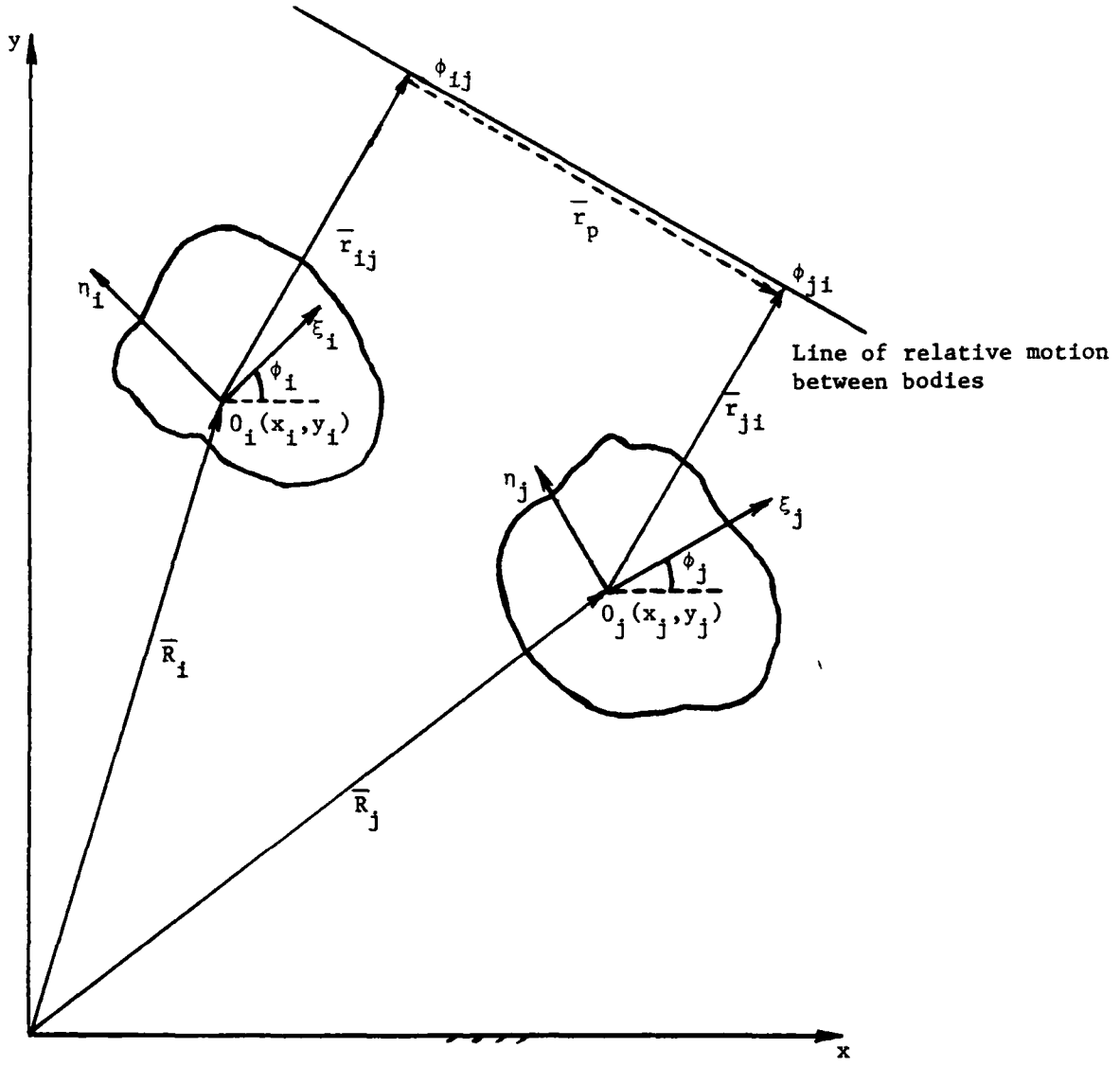
which reduces to

$$\begin{aligned} &(\xi_{ij} \cos\phi_i - \eta_{ij} \sin\phi_i + \xi_{ji} \cos\phi_j - \eta_{ji} \sin\phi_j)(x_i - x_j) \\ &+ (\xi_{ij} \sin\phi_i + \eta_{ij} \cos\phi_i + \xi_{ji} \sin\phi_j + \eta_{ji} \cos\phi_j)(y_i - y_j) \\ &+ \xi_{ij}^2 + \eta_{ij}^2 - \xi_{ji}^2 - \eta_{ji}^2 = 0 \end{aligned} \tag{2.4.4}$$



Revolute Joint

Figure 2.2



Translational Joint

Figure 2.3

A second scalar equation is obtained by noting that $\bar{r}_{ij} \times \bar{r}_{ji} = \bar{0}$. Expansion of the cross product yields only a z component, which must be zero. This is

$$(\xi_{ij} \cos \phi_i - \eta_{ij} \sin \phi_i)(\xi_{ji} \sin \phi_j + \eta_{ji} \cos \phi_j)$$
$$- (\xi_{ji} \cos \phi_j - \eta_{ji} \sin \phi_j)(\xi_{ij} \sin \phi_i + \eta_{ij} \cos \phi_i) = 0 \quad (2.4.5)$$

Spring-Dampers: Since springs and dampers generally appear in pairs, they are incorporated into a single set of equations. If one or the other is absent, its effect is eliminated by setting that term to zero. The equations for spring-damper force and torque are

$$\bar{F}_{ij} = \left[k_{ij}(\ell_{ij} - \ell_{0_{ij}}) + c_{ij} v_{ij} + F_{0_{ij}}\right] \frac{1}{\ell_{ij}} \bar{R}_{s_{ij}} \quad (2.4.6)$$

$$T_{ij} = k_{r_{ij}} (\phi_{ij} - \phi_{0_{ij}}) + c_{r_{ij}} \dot{\phi}_{ij} + T_{0_{ij}} \quad (2.4.7)$$

where

$\bar{F}_{ij}$ is the resultant force vector $[F_{x_{ij}}, F_{y_{ij}}]^T$ in the spring-damper

$\bar{R}_{s_{ij}}$ is the vector $[\ell_{ij} \cos \alpha, \ell_{ij} \sin \alpha]^T$ between points S_{ij} and S_{ji} of a spring-damper connection on the two bodies, as in Fig. 2.4

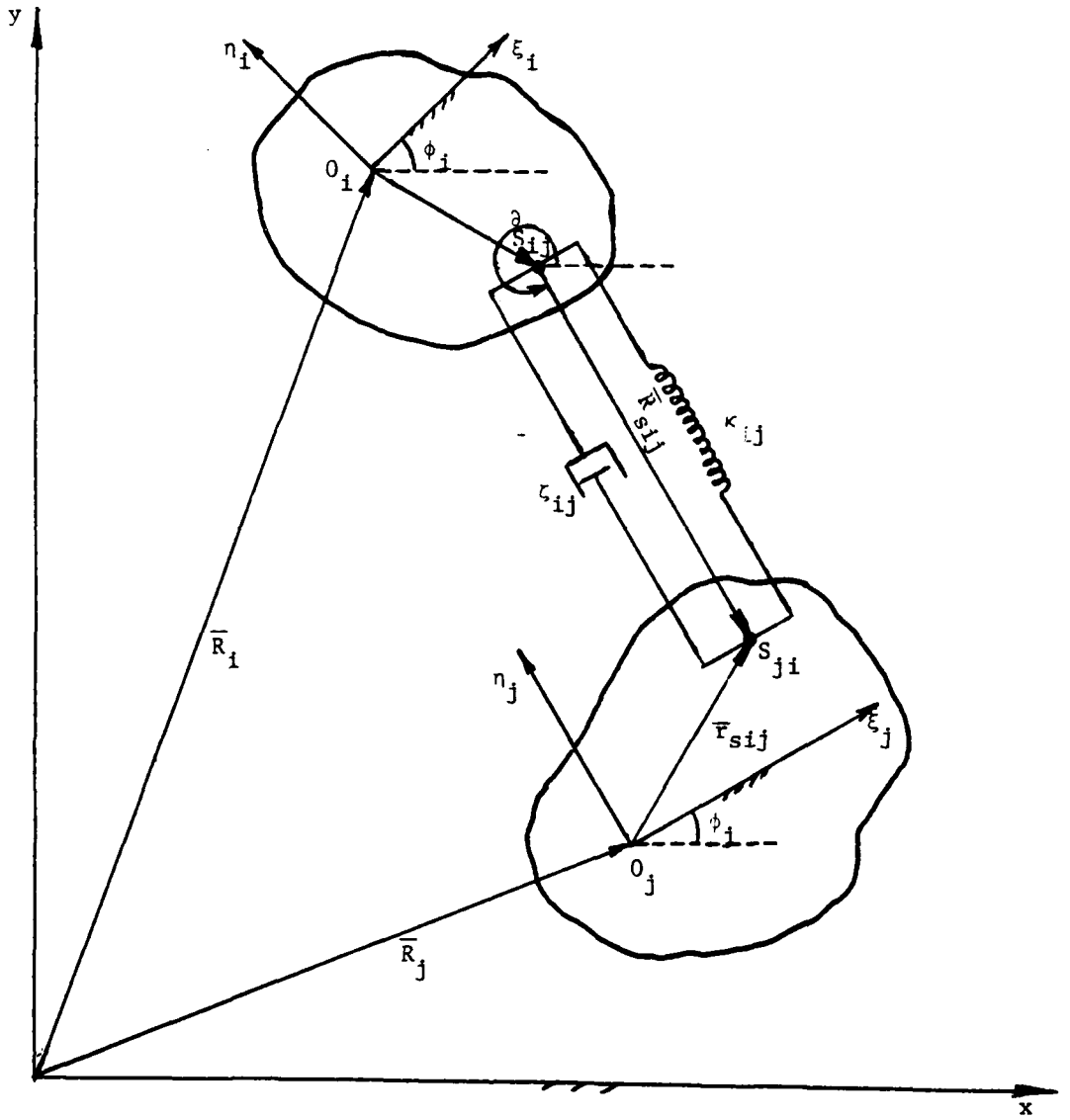
T_{ij} is the torque acting on two bodies at a revolute joint

k_{ij} and $k_{r_{ij}}$ are elastic spring coefficients

c_{ij} and $c_{r_{ij}}$ are damping coefficients

$\ell_{0_{ij}}$ and $\phi_{0_{ij}}$ are the undeformed spring length and angular rotation in the revolute

ℓ_{ij} and ϕ_{ij} are the deformed spring length and the angular rotation in the revolute, and v_{ij} and $\dot{\phi}_{ij}$ are the time derivatives of ℓ_{ij} and ϕ_{ij}



Linear Spring, Damper and Actuator

Figure 2.4

$F_{0_{ij}}$ and $T_{0_{ij}}$ are constant forces and torques applied along the spring and around the revolute joint between two bodies

From Figure 2.4, a vector expression similar to Eq. 2.4.1 is written as

$$\bar{R}_i + \bar{r}_{s_{ij}} + \bar{R}_{s_{ij}} - \bar{r}_{s_{ji}} - \bar{R}_j = 0$$

or in component form

$$\bar{R}_{s_{ij}} = \begin{bmatrix} \ell_{ij} \cos \alpha \\ \ell_{ij} \sin \alpha \end{bmatrix} = - \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} \xi_{s_{ij}} \cos \phi_i - \eta_{s_{ij}} \sin \phi_i \\ \xi_{s_{ij}} \sin \phi_i + \eta_{s_{ij}} \cos \phi_i \end{bmatrix}$$

$$+ \begin{bmatrix} x_j \\ y_j \end{bmatrix} + \begin{bmatrix} \xi_{s_{ji}} \cos \phi_j - \eta_{s_{ji}} \sin \phi_j \\ \xi_{s_{ji}} \sin \phi_j + \eta_{s_{ji}} \cos \phi_j \end{bmatrix} \tag{2.4.8}$$

where α is the angle between $\bar{R}_{s_{ij}}$ and the inertial x axis. Equation 2.4.8 is used to obtain ℓ_{ij} by noting that

$$\begin{aligned} \ell_{ij} &= [(\ell_{ij} \cos \alpha)^2 + (\ell_{ij} \sin \alpha)^2]^{1/2} \\ &= [(-x_i - \xi_{s_{ij}} \cos \phi_i + \eta_{s_{ij}} \sin \phi_i + x_j + \xi_{s_{ji}} \cos \phi_j \\ &\quad - \eta_{s_{ji}} \sin \phi_j)^2 + (- y_i - \xi_{s_{ij}} \sin \phi_i - \eta_{s_{ij}} \cos \phi_i \\ &\quad + y_j + \xi_{s_{ji}} \sin \phi_j + \eta_{s_{ji}} \cos \phi_j)^2]^{1/2} \end{aligned} \tag{2.4.9}$$

Substituting the left side of Eq. 2.4.8 into Eq. 2.4.6 the following force expressions are obtained

$$F_{x_{ij}} = |\bar{F}_{ij}| \cos \alpha \tag{2.4.10}$$

$$F_{y_{ij}} = |\bar{F}_{ij}| \sin \alpha \tag{2.4.11}$$

where cos α and sin α are obtained by dividing Eq. 2.4.8 by ℓ_{ij}. Finally, defining

$$v_{ij} = \dot{\ell}_{ij} \tag{2.4.12}$$

and transfering ℓ_{ij}, $F_{x_{ij}}$, $F_{y_{ij}}$, and v_{ij} to the right hand sides of Eqs. 2.4.9 and 2.4.12, one obtains equations in the form required by the numerical integration algorithm.

CHAPTER 3

Proposed Method of Analysis

There exist two broad categories of analysis, for which the methods of solutions are radically different. These are:

(a) kinematically driven systems.

(b) force driven systems.

In a kinematically driven system, the input is a kinematic relationship that, with the equations of constraint, specifies q. For instance, a particular generalized coordinate specified as a function of time could constitute a kinematic input. As time varies, the function changes its value. Thus, the system assumes a new configuration. In such a system, one assumes implicitly that the forces necessary to effect such an input are available.

In such a system, as will be demonstrated later, the equations of equilibrium and equations of constraint are completely decoupled and the generalized coordinates can be computed directly from the constraint equations. Once these have been computed, velocities, accelerations, and Lagrange Multipliers can be computed algebraically.

In a force driven system, forces are used as inputs. Since the effect of these forces on the generalized coordinates are not known a priori, the equations of equilibrium are coupled with the constraint equations so that they have to be solved simultaneously to yield the Lagrange Multipliers and the state variables.

Analysis of Kinematically-Determined Systems

3.1 Solution for State Variables:

The analysis scheme presented here is valid for kinematically determinate systems only. For a system with n bodies and m joints, one may write 2m

equations of constraint from equations (2.4.2), (2.4.4), and (2.4.5). There are, thus

$$f = 3n - 2m \tag{3.1.1}$$

kinematic degrees of freedom, since the constraint equations are independent. Denoting the generalized coordinates by the vector q, $q \in R^{3n}$, one has the vector of constraint equations

$$\phi_1(q) = 0, \ \phi_1 \in R^{2m}$$

In kinematic analysis, one may specify ℓ relationships among the generalized coordinates, in terms of an input parameter α

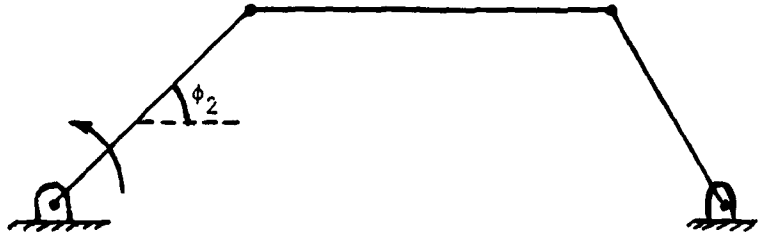
Thus, one has ℓ additional independent constraints of the form

$$\phi_2(q, \alpha) = 0, \ \phi_2 \in R^{\ell} \tag{3.1.2}$$

Since for a kinematically determinate system,

$$\ell + 2m = 3n$$

one has a situation in which the number of equations is exactly equal to the number of variables. Quite often, the parameter α is scalar time.



Crank-Rocker Mechanism

Figure 3.1

As an illustration, consider the 4-bar crank-rocker mechanism shown in Figure 3.10 that is used as the oscillating mechanism in fans. It consists

of 4 bodies, 4 revolute joints and one input relationship

$$\phi_2 = \phi_2(\alpha) \tag{3.1.3}$$

Body 1 is fixed to the Newtonian Reference Frame.

Number of generalized coordinates = 4 x 3 = 12

Number of constraints due to joints = 4 x 2 = 8

Number of constraints introduced by fixing body 1 = 3

Number of input relationships = 1

The input relationship can be thought of as a constraint, since it actually constrains the rotation of body 2. Thus, one has 12 variables and 12 constraints involving the variables. There is here one input parameter that can be varied continuously to change the configuration of the system.

Letting the vector $\phi \in R^{3n}$ represent the composite set of equations, one can write

$$\phi(q,\alpha) = \begin{bmatrix} \phi_1(q) \\ \\ \phi_2(q,\alpha) \end{bmatrix} = \begin{bmatrix} 0 \\ \\ 0 \end{bmatrix} \tag{3.1.4}$$

From the implicit function theorem of calculus, the matrix

$$\left(\frac{\partial \phi}{\partial q}\right) = \left(\frac{\partial \phi_i}{\partial q_j}\right)_{3n \times 3n}$$

is required to be non-singular for a solution to exist.

The set of 3n equations (3.1.4) is solved by the Newton Raphson for q, once α is fixed. Given an initial estimate q^i, it is required to find Δq^i, such that $q^{i+1} = q^i + \Delta q^i$ is a better approximate solution of equation (3.1.4), written now as

$$\phi(q^{i+1}, \alpha^*) = 0 \tag{3.1.5}$$

the asterisk being a reminder that the value of α is fixed.

From the Taylor expansion of (3.1.5), upto first order, one has

$$\phi(q^{i+1}, \alpha^*) = \phi(q^i, \alpha^*) + \left(\frac{\partial \phi(q^i, \alpha^*)}{\partial q}\right) q^i = 0 \qquad (3.1.6)$$

or

$$\left(\frac{\partial \phi(q^i, \alpha^*)}{\partial q}\right) \Delta q^i = - \phi(q^i, \alpha^*) \qquad (3.1.7)$$

This equation is linear in Δq^i and can be conveniently solved by using an LU factorization scheme. Let

$$\left(\frac{\partial \phi}{\partial q} (q^i, \alpha^*)\right) = LU$$

Therefore,

$$(L)(U)\Delta q^i = - \phi(q^i, \alpha^*) \qquad (3.1.8)$$

Let

$$(U)\ \Delta q^i = \nu, \ \nu \in R^{3n} \qquad (3.1.9)$$

Then,

$$L\nu = - \phi(q^i, \alpha^*) \qquad (3.1.10)$$

The vector ν can be easily determined by a forward substitution procedure. Replacing the computed values for ν in (3.1.9), Δq^i is computed by a backward substitution and the new estimate q^{i+1} is found. This iteration process is repeated until Δq^i satisfies the error tolerance specified.

3.2 Computation of Velocities:

The input parameter α may be related to time, and in some cases it may be time itself. In such a case, the generalized velocity can be easily computed.

Differentiating the constraint equations (3.1.4) with respect to time, one has

$$\left(\frac{\partial \phi}{\partial q}\right)\dot{q} + \left(\frac{\partial \phi}{\partial t}\right) = 0 \qquad (3.2.1)$$

where $(\partial\phi/\partial q) \in R^{3n}$. The first 2m terms of $(\partial\phi/\partial t)$ are always zero, since joint constraints are scheleronomic.

The L and U factors of $(\partial\phi/\partial q)$ were determined while solving for q, so all that needs to be done in computing $\dot{q}$ is forward and backward substitutions, with a different right hand side.

3.3 Computation of Accelerations:

Differentiating equation (3.2.1) with respect to time, with $\dot{q} = q(t)$ known at each time, in a fixed grid, one obtains

$$\frac{\partial}{\partial q}\left(\frac{\partial\phi}{\partial q}\dot{q}\right)\dot{q} + \frac{\partial}{\partial t}\left(\frac{\partial\phi}{\partial q}\dot{q}\right) + \frac{\partial}{\partial q}\left(\frac{\partial\phi}{\partial t}\right)\dot{q} + \frac{\partial}{\partial t}\left(\frac{\partial\phi}{\partial t}\right) = 0 \quad (3.3.1)$$

or

$$\left(\frac{\partial^2\phi}{\partial q^2}\right)\dot{q}^2 + \left(\frac{\partial^2\phi}{\partial q\partial t}\right)\dot{q} + \left(\frac{\partial\phi}{\partial q}\right)\ddot{q} + \left(\frac{\partial^2\phi}{\partial q\partial t}\right)\dot{q} + \frac{\partial^2\phi}{\partial t^2} = 0 \quad (3.3.2)$$

Rearranging (3.3.2), one has

$$\frac{\partial\phi}{\partial q}\ddot{q} = -\left\{\frac{\partial^2\phi}{\partial t^2} + \left(\frac{\partial^2\phi}{\partial q^2}\right)\dot{q}^2 + 2\left(\frac{\partial^2\phi}{\partial q\partial t}\right)\dot{q}\right\} \quad (3.3.3)$$

Once again, by a backward and forward substitution, $\ddot{q}$ can be easily determined using the L and U factors of $(\partial\phi/\partial q)$ which are already known.

3.4 Computation of the Lagrange Multipliers:

At this stage, the information available includes the present values of q, $\dot{q}$, and $\ddot{q}$. Rearranging (2.1.5), one has

$$\sum_{j=1}^{2m}\frac{\partial\phi_j}{\partial q_i}\lambda j = Q_i + \frac{\partial KE}{\partial q_i} - \frac{d}{dt}\left(\frac{\partial KE}{\partial\dot{q}_i}\right), \quad i = 1\ldots3n$$

$$= g(Q,q,\dot{q},\ddot{q}),\ g \in R^{3n} \quad (3.4.1)$$

In matrix notation, (3.4.1) simplifies to

$$\left(\frac{\partial \phi}{\partial q}\right)^T \lambda = g(Q,q,\dot{q},\ddot{q}) \tag{3.4.2}$$

Since $(\partial\phi/\partial q) = LU$, one notes that

$$\left(\frac{\partial \phi}{\partial q}\right)^T = (LU)^T = U^T L^T \tag{3.4.3}$$

Since L and U are known, L^T and U^T are also known and thus (3.4.2), which is linear in λ, can be solved to give the Lagrange Multipliers.

3.5 Analysis of Systems with Springs, Dampers and Actuator Forces:

Inclusion of springs, dampers and actuator forces in a kinematically driven system does not pose any additional computational problems. Since the values of the state variables q, $\dot{q}$, and $\ddot{q}$ are governed completely by the constraint equations, they are obviously not affected by the inclusion of springs and dampers. The Lagrange Multipliers, however, do change with the inclusion of springs and actuators, since they introduce internal forces into the system.

Since one already knows the state variables before computing the Lagrange Multipliers, the forces due to springs, dampers, and actuators can be computed for each analysis step. Referring specifically to (3.4.1), the Q_i, which originally might have consisted of constant forces, now become functions of the state variables. Thus, only Q in the right hand side of (3.4.1) needs to be modified to include spring-damper effects. The computational procedure need not change at all.

CHAPTER 4

Equilibrium in Force Driven Systems

The analysis of forced driven systems essentially follows the same procedure as transient dynamic analysis. Since the effect of the force inputs on the generalized coordinates cannot be determined without solving the equilibrium equations, the system of constraint equations and equilibrium equations are coupled and must be solved simultaneously.

However, there is one major difference in the procedures adopted for transient dynamic analysis and for equilibrium of force driven kinematic anlysis. The difference is that there are no differential equations to solve in the equilibrium problem. In kinematic analysis, one assumes that initial forces and velocity effects are negligible; hence, the equilibrium equations are algebraic. This in itself may seem to be only an academic difference, but the fact is that the numerical algorithm becomes much more stable and efficient. This is because a system of simultaneous algebraic and differential equations tends to behave like a set of stiff differential equations. Because of this, the integration algorithm is forced to take small time steps to obtain convergence of the solution in each period of time.

Once the inertial and velocity effects are removed the apparent stiffness disappears, because what remains is an algebraic nonlinear set of equations.

Springs and dampers must, however, be treated differently in this formulation, as compared to a kinematically driven system. The effect of springs and dampers are not known a priori, since the state variables and Lagrange Multipliers are computed simultaneously. One possible way to alleviate this problem is to define each spring force as a generalized coordinate. Corresponding to each spring force, a 'constraint' equation

defining it in terms of the other generalized coordinates, is added to the Jacobian matrix.

In the method presented here, the equations of equilibrium are modified to include the yet undetermined effects of the spring forces, and during the Newton iterations all state variables and unknown multipliers are simultaneously computed.

Neglecting inertial and velocity effects from (2.1.5), one has the equilibrium equations

$$\sum_{j=1}^{2m} \frac{\partial \phi_j}{\partial q_i} \lambda^j = Q_i \qquad i=1\ldots3n \tag{4.1.1}$$

the Q_i representing all external forces as well as the internal forces generated by springs and actuator forces. Since velocity effects are neglected, dampers are not considered in this formulation.

Equations (4.1.1) must be solved in conjunction with the constraint equations

$$\phi(q) = 0 \tag{4.1.2}$$

Equation (4.1.1) is linear in the Lagrange multipliers, but both equations (4.1.1) and (4.1.2) are nonlinear in the generalized coordinate q, hence, an iterative technique is needed to obtain a solution of the system of algebraic equations. Once again, the Newton-Raphson iteration scheme is used. At each analysis step of this iteration scheme, one solves a linearized version of equations (4.1.1) and (4.1.2). Linearizing these equations, one obtains

$$\frac{\partial}{\partial q}\left[\left(\frac{\partial}{\partial q}(\phi^T\lambda) - Q\right)\Big|_{q^i,\lambda^i}\right]\Delta q^i + \frac{\partial}{\partial \lambda}\left[\left(\frac{\partial}{\partial q}(\phi^T\lambda) - Q\right)\Big|_{q^i,\lambda^i}\right]\Delta\lambda^i$$

$$= -\left[\left(\frac{\partial}{\partial q}(\phi^T\lambda) - Q\right)\Big|_{q^i,\lambda^i}\right] \tag{4.1.3}$$

and

$$\left(\frac{\partial \phi}{\partial q}\right)\Big|_{q^i} \Delta q^i = -\phi(q^i) \tag{4.1.4}$$

Rewriting equations (4.1.3) and (4.1.4) in a compact matrix form, one has

$$\begin{bmatrix} \left[\frac{\partial^2}{\partial q^2}\left(\phi^T\lambda\right) - \frac{\partial Q}{\partial q}\right]\Big|_{q^i} & \left(\frac{\partial \phi}{\partial q}\right)^T\Big|_{q^i} \\ \left(\frac{\partial \phi}{\partial q}\right)\Big|_{q^i} & 0 \end{bmatrix} \begin{bmatrix} \Delta q^i \\ \Delta \lambda^i \end{bmatrix} = \begin{bmatrix} \left[\left(\frac{\partial \phi}{\partial q}\Big|_{q^i}\right)^T \lambda - Q\Big|_{q^i}\right] \\ -\phi(q^i) \end{bmatrix} \tag{4.1.5}$$

$$\left.\begin{aligned} q^{i+1} &= q^i + \Delta q^i \\ \lambda^{i+1} &= \lambda^i + \Delta \lambda^i \end{aligned}\right\} \tag{4.1.6}$$

Using an initial estimate of the q's and λ's, one solves equation (4.1.5) to obtain the optimal increments in the variables and substitutes them into (4.1.6) to generate the updated values. Iterations are continued by solving (4.1.5) again and the whole process is repeated until $\Delta\lambda$ and Δq satisfy error tolerances.

In the actual computational method used in the program, the matrix of equation (4.1.5) is not evaluated at each step. Several iterations are done using the same matrix but with an updated right hand side, before the values of the entries are changed.

While this may decrease the rate of convergence to some extent, the savings in cost realized by not re-evaluating the matrix is large enough to offset the expenses due to increased number of iterations.

CHAPTER 5

Organization of the Computer Code

The DADS-2D code essentially consists of a set of subroutines, each of which performs a specific function. The program is thus modularized so as to make it as general as possible and, at the same time, make it amenable to modifications.

As input to the program, the following details have to be supplied:

(a) System Parameters: amount of diagnostic information to be printed, error tolerances, time of simulation, the α grid interval, and the system of units to be used.

(b) Initial Estimates: of the x, y, and ϕ coordinates of each body fixed frame of reference with respect to the global frame of reference.

(c) Body Information: including mass, moment of inertia about the center of mass, and constant forces and torques acting at the body center of mass.

(d) Joint Information: which includes the bodies connected by each joint and the coordinates of the joint relative to the body-fixed axes on the connected bodies.

(e) Spring Information: consisting of the two bodies to which each spring is attached, attachment points on the two bodies, spring constants and free lengths of the springs.

In addition, provision is made to specify curves as a discrete set of points. The program then constructs a third-order spline through these points to generate an analytical function.

The main computational flow of the program is shown in Figure (5.1.1) and a detailed list of subroutines used is shown in Figure (5.1.2). A detailed list of the various subroutines used and a brief explanation of their functions is listed below.

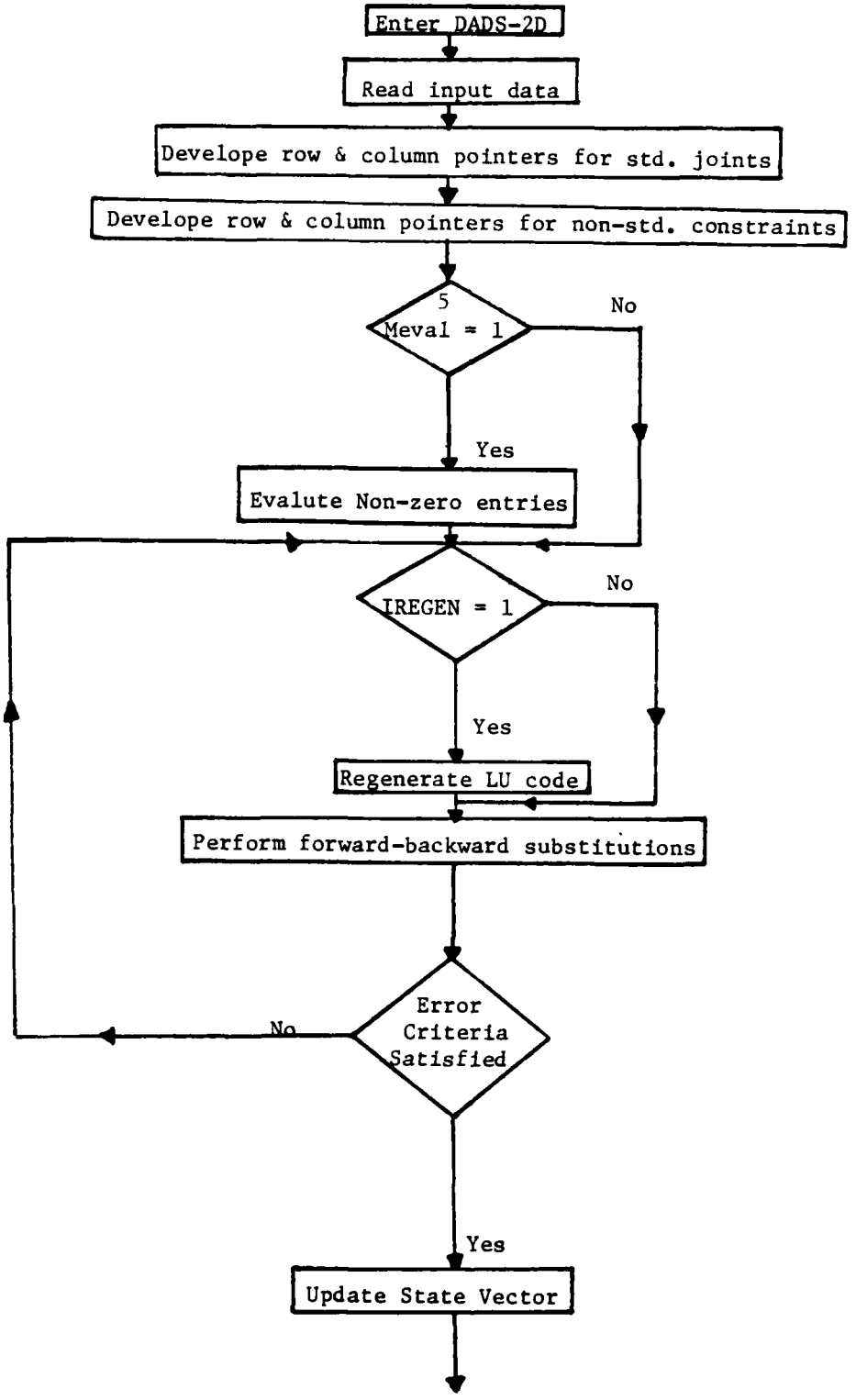
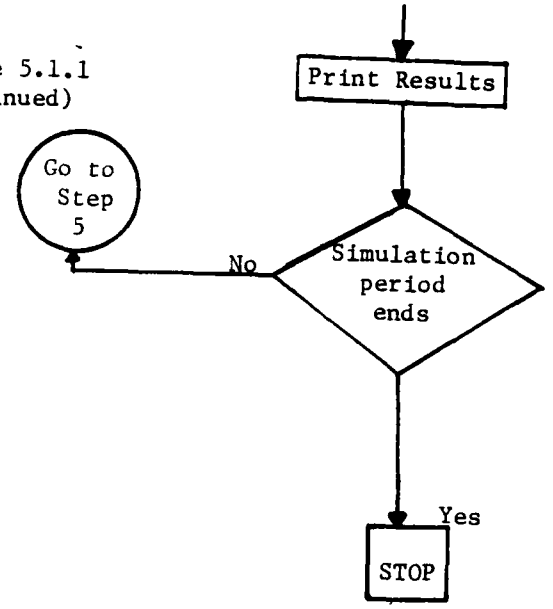


Figure 5.1.1 Flow Chart of Computer Code

Figure 5.1.1
(continued)



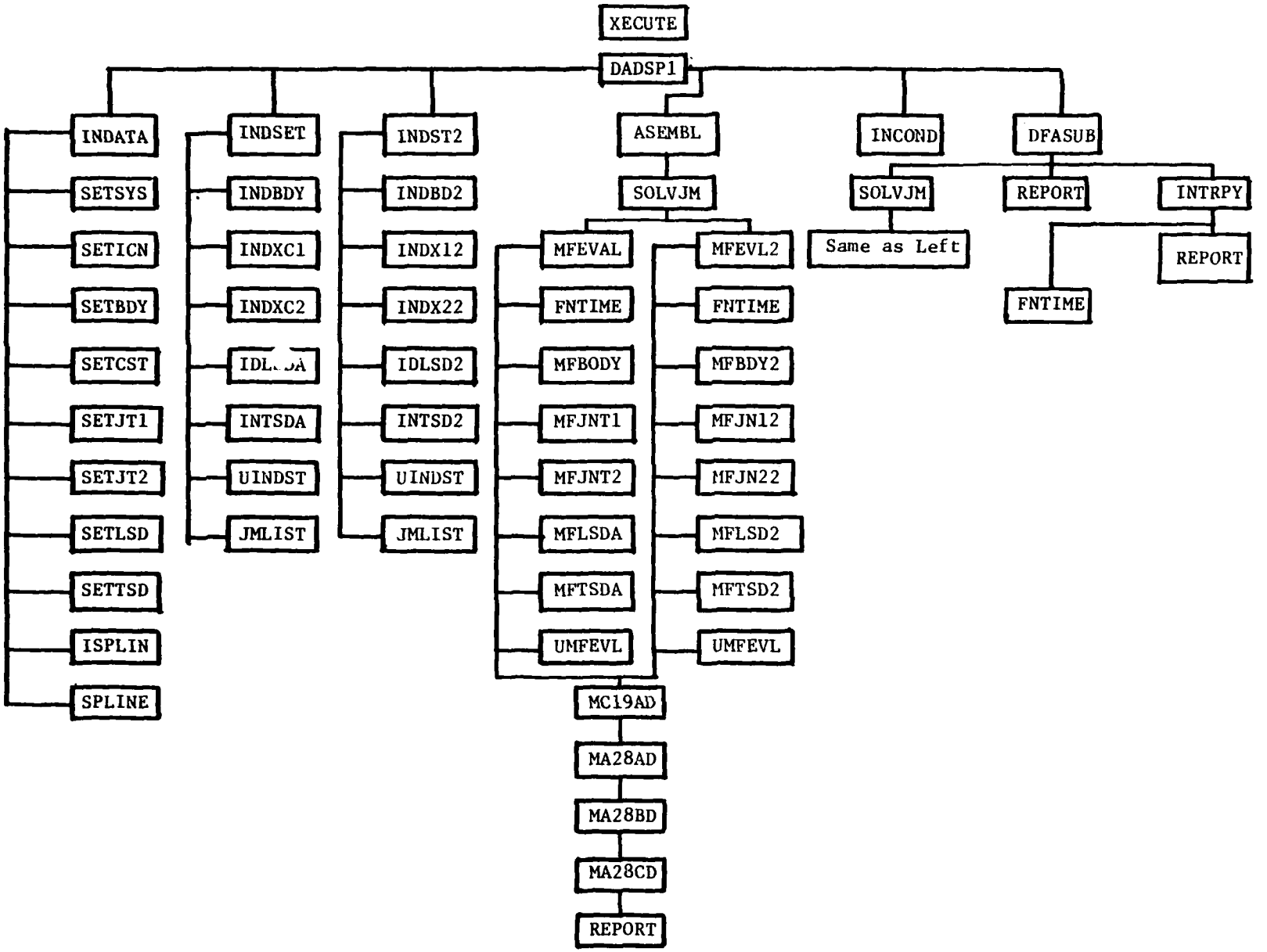


Figure 5.1.2 DETAILED FLOW CHART OF THE COMPUTER CODE

	SUBROUTINE	FUNCTION
1)	XECUTE	Calculates dimensions of the main arrays
2)	DADSP1	Main driving subroutine. Controls the entire program by calling the other subroutines
3)	INDATA	Secondary driving subroutine, calls all the 'SET' subroutines which read all the input data
4)	SETJT1, SETJT2, SETJT3, SETJT4	Read input data corresponding to revolute joints (JT1), translational joint (JT2), rev-rev joint (JT3), and rev-tran joint (JT4), respectively
5)	INDSET	Secondary driving subroutine, calls all INDX subroutines which generate the row and column indices of the non-zero entries in the Jacobian
6)	INDXC1, INDXC2 INDXC3, INDXC4	Generate row and column pointers corresponding to the constraints introduced by Joints type 1,2,3, and 4, and also the equations of motion
7)	UINDST	Generates row and column pointers for the user supplied constraints
8)	ASEMBL	Corrects input data to satisfy the constraint equations and equations of motion

	SUBROUTINE	FUNCTION
9)	MFEVAL	Secondary driving subroutine, calls other MFJ... subroutines which evaluate the non-zero entries in the Jacobian matrix
10)	MFJNT1, MFJNT2 MFJNT3, MFJNT4	Evaluate the non-zero entries introduced by the constraints due to the joint types 1,2,3, and 4.
11)	UMFEVL	Evaluates non-zero entries due to user supplied constraints
12)	SOLVJM	Secondary driving subroutine, calls the sparse matrix subroutines, performs forward and backward substitutions in the Newton Raphson iterations
13)	MC19AD, MC19BD, MA13AD MC29AD, MC29BD, MC29CD MA30AD	Sparse matrix subroutines
14)	DFASUB	The subroutine which controls all numerical computations, checks if error criteria are satisfied, predicts values at next time step by extrapolation techniques, prints the results
15)	EQUIL	Calculates the Q vector used for solving the Lagrange multipliers
16)	SPLINE {FUNCTION}	Given a set of data points, the function calculates a cubic spline through them. In addition, it computes its first and second derivatives

CHAPTER 6

Example Problems

The purpose of this section is to demonstrate the feasibility of the preceding method by solving example problems. Three kinematically determinate systems are analyzed, each of increasing complexity, and numerical results are shown. In addition, a simple force driven mechanism was analyzed.

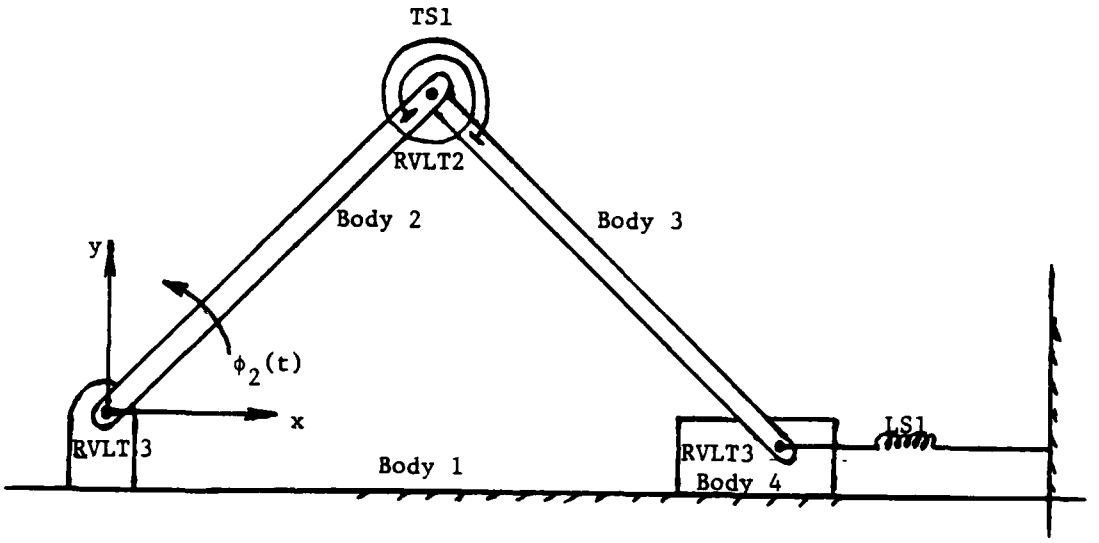
6.1 Sample Problem 1: Slider Crank

The slider crank has been chosen as the first example because it incorporates all the standardized elements of the code. Body 1 is the fixed body or 'earth'. Body 2 is the crank, body 3 is the coupler link and body 4 is the slider, which slides on a fixed guide on body 1. Crank member 2 is given an input

$$\phi_2(t) = \frac{\Pi}{4} + t/2.00$$

where t represents time. Attached to revolute joint RVLT2 is a torsional spring TS1, and between bodies 4 and 1, a linear spring LS1. It is attempted to compute the generalized coordinate and the Lagrange multipliers as functions of time. The input data for the problem are shown in Table 6.1.1.

The simulation was done using an ITEL AS-8 computer. All computations were done in double precision and the program was run using a WATFIV compiler. The simulation period was 10.0 seconds and a solution was forced every 0.1 seconds. Compiling time was 3.43 seconds, execution time was 53.29 seconds, and the total cost of the run was $18.52.



Slider Crank Mechanism

Figure 6.1.1

Body Data:

		Coordinates of Center of Mass				
Body #	ℓ	x	y	ϕ	Mass	MI
1	0	0	0	0	15.0	20.0
2	14.14	5.0	5.0	$\pi/4$	15.0	20.0
3	21.21	15.0	5.0	$-\pi/4$	20.0	25.0
4	-	25.0	-5.0	0	5.0	2.0

Spring Data:

Spring #	Spring Type	Body i	Body j	ℓ_o	θ_o	k
1	Linear	4	1	10	-	10.5
2	Torsional	2	3	-	0.1	4.5

Joint Data:

Joint #	Type	Body i	Body j	XIJ	YIJ	XJI	YJI
1	RVLT	1	2	0.0	0.0	- 7.071	0.0
2	RVLT	2	3	7.071	0.0	- 7.071	0.0
3	RVLT	3	4	14.14	0.0	0.0	0.0
4	TRAN	4	1	0.0	-1.0	0.0	- 6.0

Data for Slider-Crank Mechanism

Table 6.1.1

Time (sec)	ϕ_2(rads)	ϕ_3(rads)	X_4	LM2X	LM2Y	LM3X	Moment at Crank 2
0.0	0.78539	-0.7854	24.995				
0.5	1.03539	-0.9426	19.678	-0.5587D02	-0.9897D02	-0.5587D02	0.6083D04
1.0	1.28539	-1.0663	14.237	-0.1130D03	0.6347D03	-0.1130D03	0.5031D04
1.5	1.53539	-1.1243	9.6585	-0.1610D03	0.7667D03	-0.1610D03	0.2792D04
2.0	1.78539	-1.0910	6.7786	-0.1913D03	0.7980D03	-0.1913D03	-0.4736D03
2.5	2.03539	-0.9822	5.4396	-0.2053D03	0.7378D03	-0.2153D03	-0.3593D04
3.0	2.28534	-0.8320	5.0160	-0.2098D03	0.6603D03	-0.2098D03	-0.6099D04
3.5	2.53539	-0.6631	5.0947	-0.2090D03	0.5930D03	-6.2090D03	-0.7998D04
4.0	2.78539	-0.4872	5.5174	-0.2048D03	0.5383D03	-0.2048D03	-0.9307D04
4.5	3.03539	-0.31141	6.1295	-0.1981D03	0.4934D03	-0.1981D03	-0.1001D05
5.0	3.28539	-0.1406	7.0064	-0.1889D03	0.4564D03	-0.1889D03	-0.1013D05
5.5	3.53539	+0.0200	8.1481	-0.1769D03	0.4261D03	-0.1769D03	-0.9660D04
6.0	3.78539	0.1651	9.612	-0.1615D03	0.4028D03	-0.1615D03	-0.8640D04
6.5	4.03539	0.2878	11.4794	-0.1419D03	0.3877D03	-0.1419D03	-0.7126D04
7.0	4.28539	0.38014	13.8400	-0.1171D03	-0.3830D03	-0.11718D03	-0.5147D04
7.5	4.53539	0.4339	16.754	-0.8658D02	0.3897D03	-0.8658D02	-0.2758D04
8.0	4.78539	0.4435	20.189	-6.5051D02	0.4059D03	-0.5081D02	-0.2575D02
8.5	5.03539	0.4076	23.960	-0.1091D02	0.4253D03	-0.1091D02	0.2865D04
9.0	5.28539	0.3304	27.728	0.2865D02	0.4399D03	0.2865D02	0.5584D04
9.5	5.53539	0.2165	30.9687	0.6372D02	0.4443D03	0.6372D02	0.7784D04
10.0	5.78539	0.082681	33.5617	0.8989D02	0.4376D03	0.8989D02	0.90671D04

LM2X $\equiv$ 1st Lagrange Multiplier at revolute joint 2; LM3Y $\equiv$ 2nd Lagrange Multiplier at revolute joint 3.

TABLE 6.1.2

'EQMX	02	R001	0												
'EQMY	02	R002		1											
'EQMPHI	02	R003			2										
'XACC	02	R004	3		4	7		8							
'YACC	02	R005		5	6		9	A							
'ALPHA	02	R006				B		C	F		G				
'XVEL	02	R007					D	E		H	I				
'YVEL	02	R008							J		K	N		C	
'OMEGA	02	R009								L	M		P	Q	
'EQMX	03	R010	V		W							R		S	
'EQMY	03	R011		X	Y								T	U	
'EQMPHI	03	R012						Z							

Figure 6.1.2: Matrix of First Derivatives
of the Constraint Equation

The rows of the matrix of Fig. 6.1.2 represent constraints as follows:

R001, R002 and R003 are constraints obtained by fixing body 1.

R004 and R005 are constraints obtained due to RVLT1.

R006 and R007 are constraints obtained due to RVLT2.

R008 and R009 are constraints obtained due to RVLT3.

R010 and R011 are constraints obtained due to TRAN1.

R012 is the kinematic constraint imposed on the rotation of the body 2.

Each of the columns represents a generalized coordinate.

Thus, element (8,7) ≡ J in Fig. 6.1.2 represents the partial derivative of R008 with respect to X_3, i.e., the partial derivative of the first constraint due to RVLT3, with respect to X_3.

6.2 Example Problem 2: Peaucellier Lipkin Exact Straight Line Mechanism

Figure 6.2.1 is a schematic diagram of the Peaucellier Lipkin Exact Straight Line Mechanism. There are 8 bodies in this mechanism and 10 revolute joints. The lengths of the links comply with the conditions:

$$AC = CB = BF = AF = a$$

$$DF = DC = b$$

$$EA = ED = C$$

The mechanism always satisfies the condition

$$(DA)(BD) = a^2 - b^2 = R^2$$

where R is an inversion constant.

Body 1 is ground. Crank link 2 rotates about the fixed axis E. The motion of the crank link 2 is given as,

$$\phi_2(t) = (t/100.00)$$

where t represents time. Point B of the mechanism translates along a vertical line q-q which is always at a distance

$$h = \frac{R^2}{2(AE)}$$

from point D.

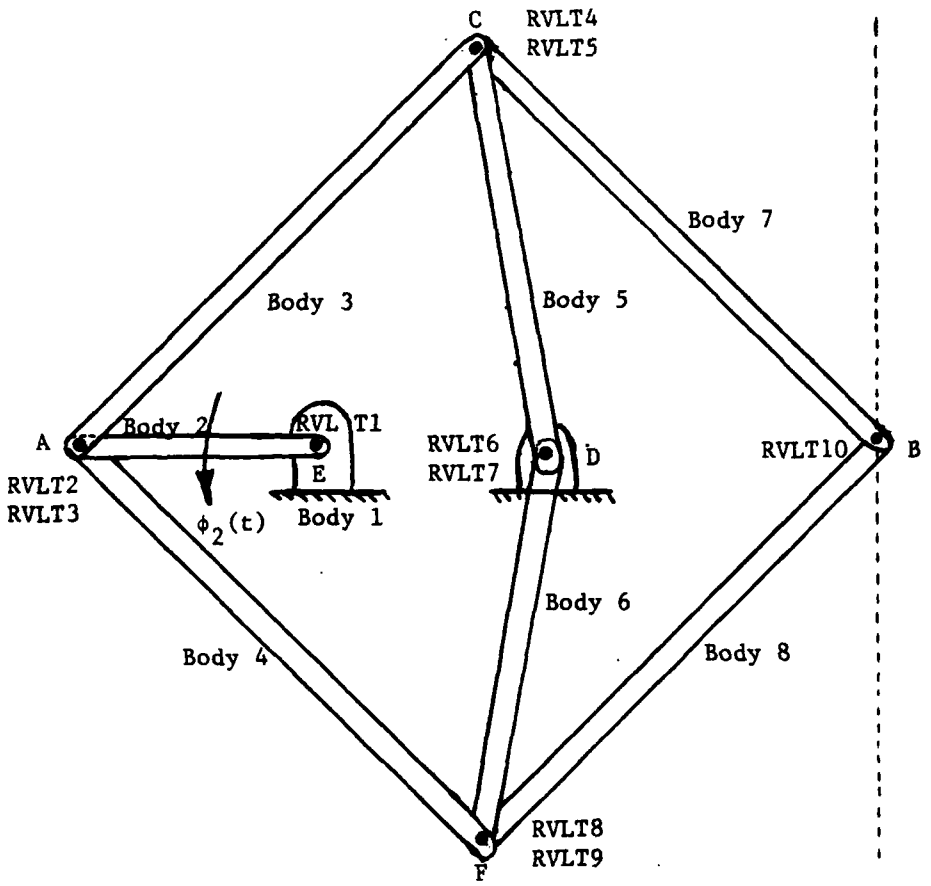


Figure 6.2.1: Peaucellier Lipkin Mechanism

BODY DATA

Body #	Length	X_{cg}	Y_{cg}	ϕ_{cg}	Mass	M. Inertia
1	5.0	0.0	0.0	0.0	0.0	0.0
2	5.0	-2.5	0.0	0.0	10.0	10.0
3	14.14	0.0	5.0	$\pi/4$	10.0	10.0
4	14.14	0.0	-5.0	$-\pi/4$	10.0	10.0
5	10.00	5.0	5.0	0.0	10.0	10.0
6	10.0	5.0	-5.0	0.0	10.0	10.0
7	14.14	15.0	0.0	$-\pi/4$	10.0	10.0
8	14.14	10.0	-5.0	$\pi/4$	10.0	10.0

JOINT DATA

Joint #	Joint Type	Body i	Body j	X1J	YIJ	XJI	YJI
1	RVLT	1	2	0.0	0.0	2.50	0.0
2	RVLT	2	3	-2.5	0.0	-7.071	0.0
3	RVLT	2	4	-2.5	0.0	-7.071	0.0
4	RVLT	3	5	7.071	0.0	-5.00	0.00
5	RVLT	3	7	7.071	0.0	-14.142	0.00
6	RVLT	5	1	-5.00	0.00	5.00	0.00
7	RVLT	6	1	5.00	0.00	5.00	0.00
8	RVLT	4	6	7.071	0.00	-5.00	0.00
9	RVLT	4	8	7.071	0.00	-7.071	0.00
10	RVLT	7	8	0.00	0.00	7.071	0.00

TABLE 6.2.1: Data for Peaucellier Lipkin Mechanism

Time (Secs)	$\phi 4$	$\phi 5$	X7	Y7	ϕ_7	LM4X
0.0	-0.78539	1.5708	15.00	0.00	0.00	0.00
0.5	-0.7603	1.5954	15.00	0.25	-0.7603	-0.721432
1.0	-0.73539	1.6195	15.00	0.00	-0.73539	-06336D02
1.5	-0.7103	1.6430	15.00	0.7514	-0.7103	-0.5220D02
2.0	-0.6853	1.6657	15.00	1.003	-0.6853	-0.6427D02
2.5	-0.6603	1.6879	15.00	1.2565	-0.6603	-0.3429D02
3.0	-0.6353	1.7095	15.00	1.5113	-0.6353	-0.2372D02
3.5	-0.6102	1.7304	15.00	1.7680	-0.6102	-0.1273D02
4.0	-0.5851	1.7506	15.00	2.0271	-0.5851	-0.1329D02
4.5	-65600	1.7702	15.00	2.2887	-05600	0.1048D02
5.0	-0.5349	1.7892	15.00	2.5534	-0.5349	0.2270D02
5.5	-0.5096	1.8074	15.00	2.82114	-0.50966	0.3533D02
6.0	-0.4843	1.8250	15.00	3.0933	-0.4843	0.4572D02
6.5	-0.4589	1.8419	15.00	3.3694	-0.4589	0.6179D02
7.0	-0.4334	1.8581	15.00	3.6502	-04.334	0.7563D02
7.5	-0.4078	1.8736	15.00	3.9362	-0.4078	0.5989D02
8.0	-0.3820	1.8883	15.00	4.2279	-0.3820	0.1045D03
8.5	-0.35604	1.9023	15.00	4.5258	-03560	0.1196D03
9.0	-0.3298	1.9155	15.00	4.8305	-0.3298	0.1351D03
9.5	-0.3034	1.9279	15.00	5.1427	-0.3039	0.1510D03

TABLE 6.2.2

At joints A, C, D, and F, three links are connected by a single revolute joint. The simulation is done by specifying two revolute joints at the same point in space, connecting the three bodies. Any revolute joint can be used to connect any two bodies, as long as one avoids having the same two bodies attached by the two revolute joints.

In the simulation model, the center of mass of link 7 is located at point B. Thus, when crank link 2 turns, one would expect the x-coodinate of link 7 to remain constant. The simulation results indeed show the above to be true.

The input data for the problem is shown in Table 6.2.1. The simulation period for this model was 10.0 seconds and a solution was forced every 0.1 seconds. Compiling time was 3.44 seconds and execution time was 25.25 seconds. All computations were done in double precision using a WATFIV compiler and an ITEL-AS-8 computer.

6.3 Example Problem 3: Link Gear Multiplier Mechanism

Figure 6.3.1 is a schematic diagram of a Link Gear Multiplier Mechanism. This is a three degrees-of-freedom model. When three inputs are prescribed (through bodies 3, 11 and 12), the output of link 6 is a product of the inputs.

There are 8 bodies, 1 revolute joint, and 10 translational joints in the mechanism. Slotted link 4 turns about the fixed axis A. Links 2, 3, 11, and 12 slide on fixed guides in body 1, and have slots in them for connections with bodies 5 and 12. When inputs $y_3(t)$, $y_{11}(t)$, and $x_{12}(t)$ are specified, the system responds with the motion

$$x_6(t) = y_3(t) \cdot x_{12}(t)/y_{11}(t)$$

Bodies 4 and 2 can turn with respect to the pin 5. Similarly, bodies 12 and 4 turn with respect to pin 8. To model this mechanism with standard elements, fictitious bodies 6, 7, 9, and 10 have been introduced. Bodies 5 and 6, 6 and 7, 8 and 9, 9 and 10 are connected by revolute joints. Body 5 slides along

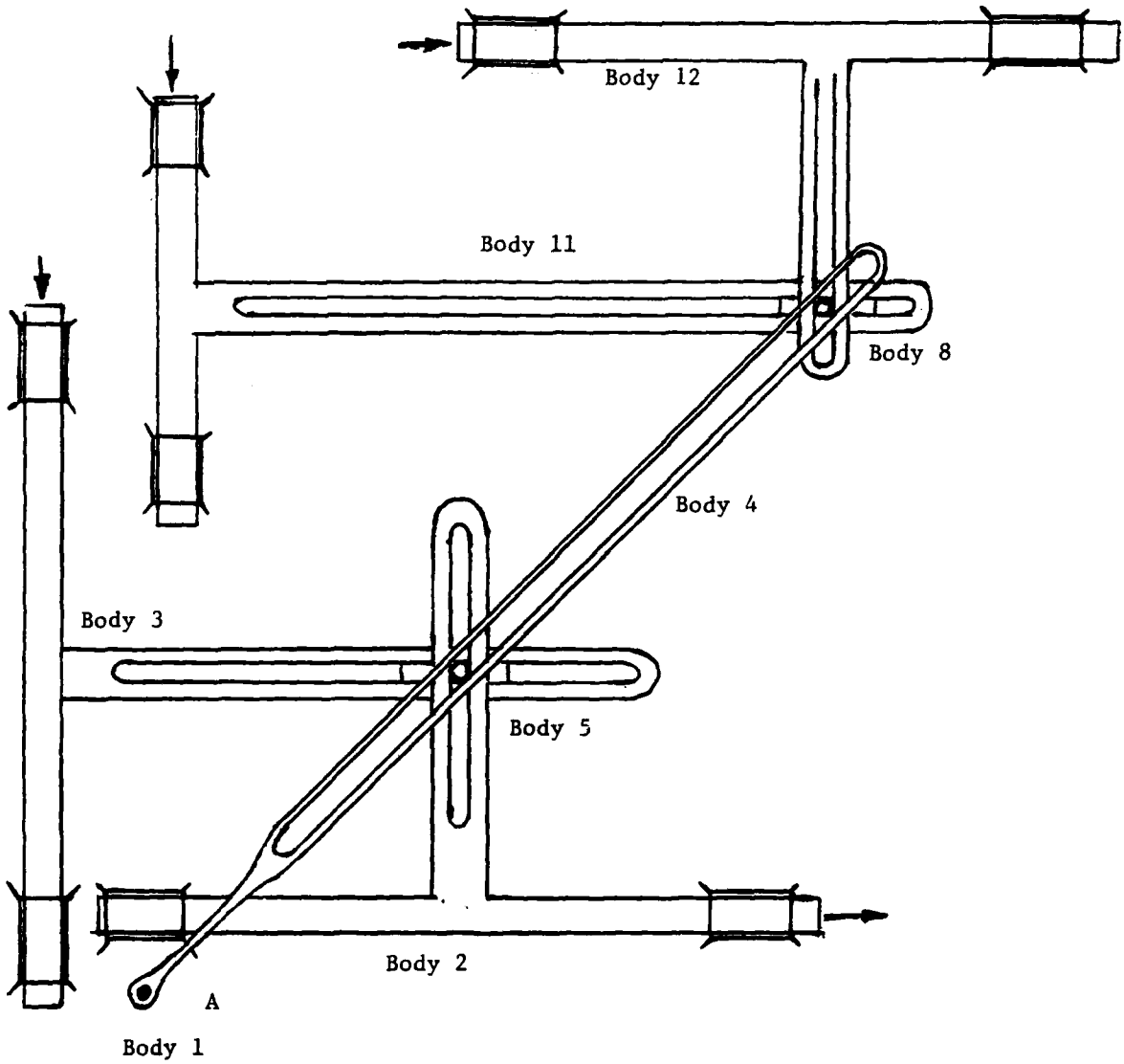


Figure 6.3.1: Link Gear Multiplier

INPUT DATA

BODY DATA

Body #	Length	X_{cg}	Y_{cg}	ϕ_{cg}	Mass	M. Inertia
1	-	0	0	0	10.00	20.00
2	-	10.00	2.00	0.00	10.00	20.00
3	-	- 2.00	10.00	0.00	10.00	20.00
4	-	5.00	5.00	0.7854	10.00	20.00
5	-	10.00	10.00	0.00	10.00	20.00
6	-	10.00	10.00	0.7854	10.00	20.00
7	-	10.00	10.00	0.00	10.00	20.00
8	-	20.00	20.00	0.00	10.00	20.00
9	-	20.00	20.00	0.7854	10.00	20.00
10	-	20.00	20.00	0.00	10.00	20.00
11	-	5.00	20.00	0.00	10.00	20.00
12	-	20.00	30.00	0.00	10.00	20.00

JOINT DATA

Type	Joint #	Body I	Body J	XIJ	YIJ	XJI	YJI
RVLT	1	1	4	0.0	0.0	-7.071	0.0
RVLT	2	5	6	0.0	0.0	0.0	0.0
RVLT	3	6	7	0.0	0.0	0.0	0.0
RVLT	4	8	9	0.0	0.0	0.0	0.0
RVLT	5	9	10	0.0	0.0	0.0	0.0
TRAN	6	1	2	0.0	1.8	0.0	-1.0
TRAN	7	1	3	-1.0	0.0	1.0	0.0
TRAN	8	1	11	2.5	0.0	-2.5	0.0
TRAN	9	1	12	0.0	25.0	0.0	-5.0
TRAN	10	3	5	0.0	- 1.0	0.0	-1.0
TRAN	11	4	6	0.0	1.0	0.0	1.0
TRAN	12	2	7	4.0	0.0	4.0	0.0
TRAN	13	11	8	0.0	- 1.0	0.0	-1.0
TRAN	14	4	9	0.0	1.0	0.0	1.0
TRAN	15	12	10	1.0	0.0	1.0	0.0

Data for Link Gear Multiplier

TABLE 6.3.1

Time (Secs)	Y3	X12	Y11	(Y3)(X12)/Y11	X6	ϕ_4
0.0	1.0000D01	2.00D01	2.0000D01	1.000D01	1.000D01	0.78539
0.5	1.0125D01	2.0250D01	2.0500D01	1.00015D01	1.00015D01	0.791533
1.0	1.0250D01	2.0500D01	2.1000D01	1.00060D01	1.006D01	0.79744
1.5	1.0375D01	2.0750D01	2.1500D01	1.00131D01	1.00131D01	0.803148
2.0	1.0500D01	2.1000D01	2.2000D01	1.00227D01	1.00227D01	0.80865
2.5	1.0625D01	2.1250D01	2.2500D01	1.00347D01	1.00347D01	0.813962
3.0	1.0750D01	2.1500D01	2.3000D01	1.00489D01	1.00489D01	0.811909
3.5	1.0875D01	2.1750D01	2.3500D01	1.00652D01	1.00652D01	0.82405
4.0	1.1000D01	2.2000D01	2.4000D01	1.00833D01	1.0083D01	0.82885
4.5	1.1125D01	2.2250D01	2.450D01	1.01033D01	1.0103D01	0.83349
5.0	1.1250D01	2.2500D01	2.500D01	1.01250D01	1.0125D01	0.83798
5.5	1.1375D01	2.2750D01	2.5500D01	1.01480D01	1.01483D01	0.84233
6.0	1.1500D01	2.300D01	2.6000D01	1.01731D01	1.01731D01	0.84654
6.5	1.1625D01	2.3250D01	2.6500D01	1.01993D01	1.01993D01	0.85063
7.0	1.1750D01	2.3500D01	2.7000D01	1.0226 9D01	1.02269D01	0.85459
7.5	1.1875D01	2.3750D01	2.7500D01	1.02550D01	1.0255D01	0.85843
8.0	1.200D01	2.4000D01	2.8000D01	1.02857D01	1.02857D01	0.86217
8.5	1.2125D01	2.4250D01	2.8500D01	1.03169D01	1.0316D01	0.86579
9.0	1.2250D01	2.4500D01	2.9000D01	1.03491D01	1.03491D01	0.86931
9.5	1.2375D01	2.475D01	2.9500D01	1.03824D01	1.03824D01	0.87273

TABLE 6.3.2

body 3, body 6 slides on body 4 and body 7 slides on body 2. Similarly, body 8 slides on body 11, body 9 slides on body 4, and body 10 slides on body 12.

The model used for simulation thus has 12 bodies, 5 revolute joints and 10 translational joints. The dimensions of the mechanism may be arbitrary, provided the topological configuration does not change. The input data for the problem are shown in Table 6.3.1.

Table 6.3.2 contains results of the simulation. Columns 2, 3, and 4 contain the values Y3, X12, and Y11, respectively. Using these values, the expected value of X6 is computed and printed in column 5. Column 6 contains the actual value of X6, as obtained from the simulation.

The simulation period for the job was 10.0 seconds, with a solution being forced every 0.10 seconds. All computations were done using a WATFIV compiler, on an ITEL AS-8 computer. Compilation time was 3.51 seconds, and execution time 32.46 seconds. Total cost of the job was $20.79.

6.4 Example Problem 4: Force driven 4-bar linkage

Figure 6.4.1 is a schematic figure of a 4-bar linkage, which has one degree-of-freedom. The mechanism is initially in a non-equilibrium position. The problem at hand is to find the equilibrium position of the mechanism when it is subject to its own body forces.

Pertinent input data are shown in Table 6.4.1. The values of the generalized coordinates and the Lagrange multipliers at equilibrium position are computed.

It must be noted that for the force driven case, the program attempts to find an equilibrium position if such a position exists at all. Thus, there is no parameter like time, as in analysis of kinematically determinate systems.

BODY DATA

Body #	Length	X_{cm}	Y_{cm}	ϕ_{cm}	Mass	MI
1	100	0.0	-0004	0.0	0.0	0.0
2	100	8.682	-49.240	-1.3962	1.0	1.0
3	100	67.36	-98.480	0.0	1.0	1.0
4	100	108.682	-49.24	-1.3962	1.0	1.0

JOINT DATA

Joint #	Joint Type	Body I	Body J	X1J	Y1J	XJI	YJI
1	RVLT	1	2	0.0	0.0	-50.0	0.0
2	RVLT	2	3	50.0	0.0	-50.0	0.0
3	RVLT	3	4	50.0	0.0	-50.0	0.0
4	RVLT	4	1	50.0	0.0	-50.0	0.0

Data for Four-bar Linkage

Table 6.4.1

RESULTS

Estimates obtained using ϕ_2 = const. = -1.3962 rads

Body #	X	Y	ϕ(Rads)
1	10^{-14}	0	0
2	8.68553	-49.2398	-1.3962
3	67.3711	-98.479	10^{-14}
4	108.6855	-49.2398	1.7453

Estimates of Lagrange Multipliers

JT #	λx	λy
1	45.40	386.08
2	45.40	128.69
3	45.40	-128.69
4	45.40	-386.08

Constraint

$\phi_2 - 10^0 = 0$ 8942.34 (Moment)

Execution time = 0.51 secs. in the ITEL AS-8 Computer

# of iterations = 2

Table 6.4.2

FINAL RESULTS

Generalized Co-ordinations

Body #	X	Y	ϕ(Rads)
1	0	0	0
2	10^{-14}	-50.00	-1.5708
3	50.00	-100.00	10^{-17}
4	100.00	- 50.00	1.5708

Lagrange Multipliers

Joint #	λx	λy
1	10^{-13}	386.08
2	10^{-13}	128.696
3	10^{-13}	-128.696
4	10^{-13}	-386.088

CPU Time = 1.02 secs

# of iterations = 2

Table 6.4.3

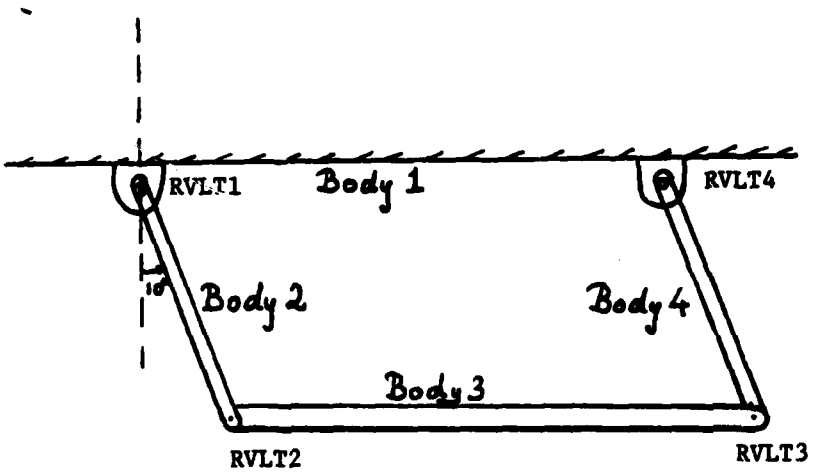


Figure 6.4.1: Four Bar Linkage

The solution strategy for force driven mechanisms is slightly different from what has been explained in the previous examples. The program has been coded such that all Lagrange multipliers have zero initial estimates. During subsequent iterations for closure of the mechanism, these estimates are expected to reach the correct values.

The difficulty with this strategy is that very often the Jacobian matrix of first derivatives does not have full row rank especially when all Lagrange multipliers have zero initial estimates. To overcome the problem caused by singularity of the matrix, the sparse matrix code fixes values for K of the state variables arbitrarily where $\overset{\searrow}{K}$ is the nullity of the matrix, and proceeds to solve for the remaining unknowns from the largest non-singular submatrix available.

Thus, the configuration of the mechanism is changed considerably from what was initially specified. Since the initial estimates of the dependent variables have not changed, the input data is not a good estimate any more. Thus while attempting loop closure, the program may need several iterations and in some cases closure may not be attained at all. To offset this computational difficulty, the following strategy is employed: If a mechanism has f degrees-of-freedom, f coordinates are kept fixed by the code at their initial estimates. Thus, one can now obtain the generalized coordinates by solving the constraint equations. Once all the variables have been obtained, Lagrange multipliers are computed.

The values computed above are read as initial estimates for the force driven system. Since one has a reasonable estimate of the Lagrange multipliers and other generalized coodinates, closure will be attained reasonably quickly.

This method may not work well for problems in which the initial estimates are inaccurate, but for reasonably good initial estimates closure is guaranteed.

CHAPTER 7

Conclusions

The basic attempt in this report is to formulate and code a general purpose method for kinematically analyzing force driven and kinematically driven mechanisms.

It was shown that for the kinematically driven systems constraint equations can be used to evaluate the generalized coordinates. Successive differentiation of the constraint equations yield velocities and accelerations. The equations of equilibrium were shown to be uncoupled from the constraint equations, and the unknown Lagrange multipliers can be computed even after including inertial effects. In all cases, it was shown that the coefficient matrix is the $(\partial\phi/\partial q)$ matrix or its transpose, and hence, computations of velocities, accelerations and Lagrange multipliers are not very expensive.

It was shown that for the force driven system, the equations of equilibrium and the constraint equations were coupled. To avoid solving the differential equations of motion, it was assumed that inertial effects were negligible and thus this solution technique is applicable only to kinetostatic-quasistatic systems.

Three example problems of varying complexity were solved, and the solution strategy was shown to be feasible. Solution of velocities and accelerations have not been included in the present code but can be done with very few modifications in the main program.

REFERENCES

1) Haug, E.J., Wehage, R., and Huang, R.C., Computational Methods in Mechanical System Dynamics, College of Engineering, The University of of Iowa, 1980.

2) Duff, I.S., MA-28 - A Set of Fortran Subroutines for Sparse Unsymmetric Linear Equations, Harwell, United Kingdom Atomic Authority, July, 1972.

3) Kaufman, R.E., "Kinematic and Dynamic Design of Mechanisms " in Shock and Vibration Computer Program Reviews and Summaries, ed. by W. & B. Pilkey, Shock and Vibration Information Center, Naval Research Lab., Code 8404, Washington, D.C., 1975.

4) Paul, Burton, Kinematics and Dynamics of Planar Machinery, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1979.

5) Uicker, J.J., "Users' Guide for IMP: A Problem-Oriented Language for the Computer-Aided Design and Analysis of Mechanisms," NSF Report, Res. Grant GK-4552, University of Wisconsin, Madison, Wisc., 1973.

6) Paul, B. and Hud, G., "Users' Manual for Program Dymac-L2", MEAM Report 76-5, Dept. of Mechanical Engg., Univ. of Pennsylvania, Phila.,1975.

7) Chace, M.A. and Smith, D.A., "DAMN-Digital Computer Program for the Dynamic Analysis of Generalized Mechanical Systems," SAE Trans., 80, 969-983, 1971.

8) Dix, R.C. and Lehman, T .J., "Simulation of the Dynamics of Machinery," J. Engg. Ind. Trans., ASME Ser. B, 94, 433-438, 1972.

9) Orlandea, N., Chace, M.A. and Calahan, D.A., "A Sparsity Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems - Parts I and II," J. Engg. Ind. Trans., ASME Ser. B, 99, 733-779, 780-784, 1977.

10) Wehage, R.A., Haug, E.J., Barman, N.C., "Dynamic Analysis and Design of Constrained Mechanical Systems," Technical Report No. 50, Contract No. DAAK30-78-C-0096, November, 1978.

11) Langrana, N.A. and Bartel, D.L., "An Automated Method for Dynamic Analysis of Spatial Linkages for Biomechanical Systems," J. Engg. Ind. Trans., ASME Ser. B, 97, 556-574, 1975.

12) Huston, R.L., Passerello, C.E. and Harlow, M.W., UCIN - Vehicle Occupant/ Crash Victim Simulation Model, Structural Mechanics Software Series, University Press of Virginia, Charlottesville, VA, 1977.

13) Goldstein, Herbert, Classical Mechanics, Addison-Wesley Publishing Company, Reading, Massachusetts, 1950.

DISTRIBUTION LIST

Please notify USATACOM, DRSTA-ZSA, Warren, Michigan 48090, of corrections and/or changes in address.

Commander (25)
US Army Tk-Autmv Command
R&D Center
Warren, MI 48090

Superintendent (02)
US Military Academy
ATTN: Dept of Engineering
Course Director for
Automotive Engineering

Commander (01)
US Army Logistic Center
Fort Lee, VA 23801

US Army Research Office (02)
P.O. Box 12211
ATTN: Dr. F. Schmiedeshoff
Dr. R. Singleton
Research Triangle Park, NC 27709

HQ, DA (01)
ATTN: DAMA-AR
Dr. Herschner
Washington, D.C. 20310

HQ, DA (01)
Office of Dep Chief of Staff
for Rsch, Dev & Acquisition
ATTN: DAMA-AR
Dr. Charles Church
Washington, D.C. 20310

HQ, DARCOM
5001 Eisenhower Ave.
ATTN: DRCDE
Dr. R.L. Haley
Alexandria, VA 22333

Director (01)
Defense Advanced Research
Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Commander (01)
US Army Combined Arms Combat
Developments Activity
ATTN: ATCA-CCC-S
Fort Leavenworth, KA 66027

Commander (01)
US Army Mobility Equipment
Research and Development Command
ATTN: DRDME-RT
Fort Belvoir, VA 22060

Director (02)
US Army Corps of Engineers
Waterways Experiment Station
P.O. Box 631
Vicksburg, MS 39180

Commander (01)
US Army Materials and Mechanics
Research Center
ATTN: Mr. Adachi
Watertown, MA 02172

Director (03)
US Army Corps of Engineers
Waterways Experiment Station
P.O. Box 631
ATTN: Mr. Nuttall
Vicksburg, MS 39180

Director (04)
US Army Cold Regions Research
& Engineering Lab
P.O. Box 282
ATTN: Dr. Freitag, Dr. W. Harrison
Dr. Liston, Library
Hanover, NH 03755

President (02)
Army Armor and Engineer Board
Fort Knox, KY 40121

Commander (01)
US Army Arctic Test Center
APO 409
Seattle, WA 98733

Commander (02)
US Army Test & Evaluation
Command
ATTN: AMSTE-BB and AMSTE-TA
Aberdeen Proving Ground, MD
21005

Commander (01)
US Army Armament Research
and Development Command
ATTN: Mr. Rubin
Dover, NJ 07801

Commander (01)
US Army Yuma Proving Ground
ATTN: STEYP-RPT
Yuma, AZ 85364

Commander (01)
US Army Natic Laboratories
ATTN: Technical Library
Natick, MA 01760

Director (01)
US Army Human Engineering Lab
ATTN: Mr. Eckels
Aberdeen Proving Ground, MD
21005

Director (02)
US Army Ballistic Research Lab
Aberdeen Proving Ground, MD
21005

Director (02)
US Army Materiel Systems
Analysis Agency
ATTN: AMXSY-CM
Aberdeen Proving Ground, MD
21005

Director (02)
Defense Documentation Center
Cameron Station
Alexandria, VA 22314

US Marine Corps (01)
Mobility & Logistics Division
Development and Ed Command
ATTN: Mr. Hickson
Quantico, VA 22134

Keweenaw Field Station (01)
Keweenaw Research Center
Rural Route 1
P.O. Box 94-D
ATTN: Dr. Sung M. Lee
Calumet, MI 49913

Naval Ship Research & (02)
Dev Center
Aviation & Surface Effects Dept
Code 161
Washington, D.C. 20034

Director (01)
National Tillage Machinery Lab
Box 792
Auburn, AL 36830

Director (02)
USDA Forest Service Equipment
Development Center
444 East Bonita Avenue
San Dimes, CA 91773

Engineering Societies (01)
Library
345 East 47th Street
New York, NY 10017

Dr. I.R. Erlich (01)
Dean for Research
Stevens Institute of Technology
Castle Point Station
Hoboken, NJ 07030

Grumman Aerospace Corp (02)
South Oyster Bay Road
ATTN: Dr. L. Karafiath
Mr. F. Markow
M/S A08/35
Bethpage, NY 11714

Dr. Bruce Liljedahl (01)
Agricultural Engineering Dept
Purdue University
Lafayette, IN 46207

Mr. H.C. Hodges (01)
Nevada Automotive Test Center
Box 234
Carson City, NV 89701

Mr. R.S. Wismer (01)
Deere & Company
Engineering Research
3300 River Drive
Moline, IL 61265

Oregon State University (01)
Library
Corvallis, OR 97331

Southwest Research Inst (01)
8500 Culebra Road
San Antonia, TX 78228

FMC Corporation (01)
Technical Library
P.O. Box 1201
San Jose, CA 95108

Mr. J. Appelblatt (01)
Director of Engineering
Cadillac Gauge Company
P.O. Box 1027
Warren, MI 48090

Chrysler Corporation (02)
Mobility Research Laboratory,
Defense Engineering
Department 6100
P.O. Box 751
Detroit, MI 48231

CALSPAN Corporation (01)
Box 235
Library
4455 Benesse Street
Buffalo, NY 14221

SEM, (01)
Forsvaretsforskningsanstalt
Avd 2
Stockholm 80, Sweden

Mr. Hedwig (02)
RU III/6
Ministry of Defense
5300 Bonn, Germany

Foreign Science & Tech (01)
Center
220 7th Street North East
ATTN: AMXST-GEI
Mr. Tim Nix
Charlottesville, VA 22901

General Research Corp (01)
7655 Old Springhouse Road
Westgate Research Park
ATTN: Mr. A. Viilu
McLean, VA 22101

Commander (01)
US Army Developmant and
Readiness Command
5001 Eisenhower Avenue
ATTN: Dr. R.S. Wiseman
Alexandria, VA 22333